



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Aplicación del Desarrollo Centrado en el Usuario para el diseño de una aplicación de gestión de una compañía de danza (Quakes).

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Cuñat Gómez, Jose Enrique

Tutor: Manuela Albert Albiol

2017-2018

Desarrollo Centrado en el Usuario para la construcción de una aplicación móvil de gestión de una compañía de danza (Quakes)

RESUMEN

Este TFG abordará el diseño e implementación de una aplicación para una compañía de danza real empleando metodologías del diseño centrado en el usuario.

El objetivo del producto es abarcar las necesidades del cliente, implementando diversas funcionalidades para cubrir problemas de gestión de la información que posee la compañía en el día a día, sobre todo, contenido multimedia y fechas importantes.

Se emplearán las técnicas y metodologías que ofrece el diseño centrado en el usuario para crear una aplicación lo más usable y personalizada al cliente. Se realizarán análisis, diversos prototipos, evaluaciones y por último la aplicación que usarán.

Para la implementación del producto se utilizará el framework Ionic 3, conjunto con Angular 2 y HTML.

Palabras clave: Diseño, Usuario, Ionic, Compañía de Danza.

ABSTRACT

This end-of-degree project will address the design and implementation of an application for a real dance company using user-centered design methodologies.

The objective of the product is to cover the needs of the client, implement various functions for the treatment of information management problems that the company has in the day to day, especially multimedia content and important dates.

The techniques and methods offered by the design in the user have been designed to create a more useful and personalized application to the client. Analysis, various prototypes, evaluations and finally the application they will use will be performed.

For the implementation of the product, the Ionic 3 framework is used, together with Angular 2 and HTML.

Keywords: Desing, User, Ionic, Dance Company.

RESUM

Aquest TFG abordarà el disseny i implementació d'una aplicació per a una companyia de dansa real emprant metodologies del disseny centrat en l'usuari.

L'objectiu del producte és abastar les necessitats del client, implementant diverses funcionalitats per cobrir problemes de gestió de la informació que posseeix la companyia en el dia a dia, sobretot, contingut multimèdia i dates importants.

Es faran servir les tècniques i metodologies que ofereix el disseny centrat en l'usuari per a crear una aplicació el més usable i personalitzada al client. Es realitzaran anàlisis, diversos prototips, avaluacions i finalment l'aplicació que faran servir.

Per a la implementació del producte es farà servir el framework Ionic 3, conjunt amb Angular 2 i HTML.

Palabras clave: Disseny, Usuari, Ionic, Companyia de Dansa.

TABLA DE CONTENIDOS

1.	Introducción	7
2.	Estado del arte	9
3.	Análisis de Necesidades de Usuario	10
3.1	Investigación cualitativa	10
3.1.1	Entrevistas	10
3.1.2	Análisis de las entrevistas.....	11
3.1.3	Cuestionarios	12
3.1.4	Análisis de los cuestionarios.....	13
3.1.5	Conclusión de la investigación cualitativa.	14
3.2	Persona Primaria y Escenarios	14
3.2.1	Persona Primaria	14
3.2.2	Escenarios.....	15
4.	Diseño y Evaluación	17
4.1	Prototipo de Baja fidelidad	17
4.1.1	Diseño del prototipo	17
4.1.2	Evaluación del prototipo	20
4.1.3	Reforma del prototipo	22
4.1.4	Evaluación del prototipo	25
4.2	Prototipo de Alta fidelidad.....	26
4.2.1	Diseño del primer prototipo de alta fidelidad	26
4.2.2	Evaluación del primer prototipo de alta fidelidad	32
4.2.3	Reforma del prototipo.....	34
4.2.4	Segunda evaluación del prototipo	36
5.	Implementación	38
5.1	Arquitectura del sistema	38
5.2	<i>SplashScreen</i> y fase de <i>Login</i>	39
5.3	Navegación: Creación del <i>TabPage</i>	40
5.4	<i>HomePage</i>	41
5.5	Calendario.....	43
5.6	Tutoriales	44
5.7	Galería	46
5.8	Música	47
5.9	Clean Code.....	48



Desarrollo Centrado en el Usuario para la construcción de una aplicación móvil de gestión de una compañía de danza (Quakes)

6.	Pruebas	49
6.1	Diseño de Pruebas.....	49
6.1.1	Vista Login	49
6.1.2	Vista Anuncios.....	50
6.1.3	Calendario.....	50
6.1.4	Galería.....	52
6.1.5	Tutoriales.....	55
6.1.6	Reproductor de Música	55
7.	Conclusión	57
7.1	Relación del trabajo desarrollado con los estudios cursados	57
	Referencias	58
	Apéndice	59
	Apéndice I – Manual de Usuario.....	59

1. Introducción

Quakes es una compañía de danza valenciana, la cual posee una cantidad de 16 bailarines con rangos y responsabilidades diferentes. A causa del incremento de demandas en el sector, ha decidido incluir en su metodología de trabajo una aplicación móvil con la cual gestionar toda la información que poseen, tener una mayor facilidad a la hora de tratar los datos que se manejan entre los integrantes de la compañía y mejorar la relación/comunicación entre los integrantes de la entidad.

La aplicación a crear necesita ofrecer una buena experiencia de usuario para poder garantizar un buen uso del producto entre los componentes de la asociación. Con el fin de poder conseguir este objetivo, se opta a aplicar una Metodología Centrada en el Usuario para su desarrollo. Utilizando técnicas y recursos del sistema a emplear obteniendo un resultado eficiente y práctico para el cliente final.

La aplicación ofrecerá unos servicios sacados de las necesidades de los integrantes, los cuales serán vistos mediante un análisis de las necesidades del cliente, de esta forma poder saber lo que realmente necesita y no lo que ellos piensan que necesitan. Ofreciendo una plataforma para poder solventar sus tareas cotidianas y realizarlas de manera más sencilla y eficiente.

Entre los futuros usuarios de la aplicación se ha observado que poseen terminales de diferentes sistemas operativos, siendo los presentes los más conocidos: iOS y Android. Por esta razón se ha decidido emplear *IONIC*¹, un *framework* gratuito el cual ofrece la posibilidad de crear aplicaciones híbridas empleando el uso de, HTML, CSS y JS. Basado en el modelo MVC (Modelo Vista Controlador) separando en tres componentes, los datos, la interfaz final y la lógica de la vista.

El TFG (trabajo de fin de grado) se estructurará en el orden que ofrece la metodología DCU (Diseño Centrada en el Usuario) centrándose más en el proceso de elaboración que en la implementación final. El trabajo posee una forma iterativa donde se emplearán de diferentes versiones prototipos y sus posteriores evaluaciones para llegar al producto de mayor eficiencia posible teniendo en todo momento presente al usuario en la parte evolutiva de la aplicación,

El alcance del trabajo incluirá la parte del *frontend*. Esta parte empleará Angular², ofrecido por el propio *framework*, respaldado por el lenguaje de *TypeScript*. Este lenguaje está basado en el empleo de objetos, y posee gran sinergia con *JavaScript*, siendo este uno de los lenguajes estudiados a lo largo de la carrera. El frontend lanzará peticiones HTTP, pero la API no estará incluida en el alcance del proyecto. Se hará referencia y posterior justificación de lo que sucederá en estas llamadas y la necesidad de hacerlas, pero se asumirá que esta API existe y estaría preparada para las llamadas de la aplicación final.

Aspectos a tener en cuenta

Se tendrá en cuenta la jerarquía interna de la organización, dividiendo entre dos tipos de rol (administrador y usuario). Estos poseerán restricciones y limitaciones diferentes, por ello se tendrán en cuenta ambos usuarios en el proceso gradual de la aplicación móvil. También se tendrá en cuenta que al tratarse de usuarios reales se tendrá un testeo y evaluación real. Se ofrecerá un manual de usuario para poder entender cómo usar la aplicación.

¹ <https://ionicframework.com/>

² <https://angular.io/>

Motivación

Se ha decidido desarrollar esta aplicación para solventar un problema de una compañía real y tener una forma más cercana de trabajar con un proyecto existente. También favorece el trato a un usuario concreto tomando datos verídicos.

Implementada con un framework mencionado a lo largo de la carrera para afianza el conocimiento de tecnologías emergentes con gran peso en el mercado laboral. Esto ayudará al alumno a necesitar de información externa a la ofrecida por la universidad para mejorar sus aptitudes a la hora de crear nuevos proyectos en un futuro y mejorar su capacidad de autoaprendizaje.

Objetivos

- Construir una aplicación móvil desde el primer paso
- Conocer y emplear las metodologías centradas en el usuario
- Testear y evaluar de una manera directa con un usuario auténtico.
- Aprendizaje en nuevas y diferentes tecnologías: *Ionic, Angular, TypeScript*.
- Solventar un problema de una asociación existente.

Estructura

Ligeramente mencionada anteriormente, se poseerá una estructura iterativa, donde tras explicar brevemente como conseguir las necesidades del usuario mediante técnicas de investigación cualitativas, se basará en el siguiente orden:

1. Diseño de una interfaz (mediante diferentes herramientas a lo largo del proceso)
 - I. En este punto se desarrollarán diferentes prototipos para el testeo directo con el usuario, pudiendo de esta forma evaluar de una forma visual y directa. Cada vez se emplearán herramientas diferentes hasta acercarse al producto final.
2. Evaluación y testeo
 - I. Mediante técnicas de evaluación se analizarán los fallos, errores y posibles mejoras del prototipo para ir estrechando el camino hasta llegar al producto final.
3. Resumen de los informes y cambios a realizar
 - I. Se analizará la evaluación sacando en claro los cambios a realizar en caso de que los haya. Si los hubiera, se volvería a realizar una iteración comenzando de nuevo por el primer punto hasta conseguir la aplicación deseada e implementada.

2. Estado del arte

Se ha analizado el campo en el que se engloba el producto que se va a desarrollar. No se ha encontrado una aplicación que sea igual a la que se pretende implementar, en cambio, sí que se ha logrado acceder a apps móvil e incluso alguna app web que posean similitudes o posibles futuras funciones que pueda poseer nuestra aplicación.

Sobre todo, las aplicaciones que actualmente hay en el mercado activas, son aplicaciones que gestionan las necesidades que pueda poseer una academia concreta, dando facilidades, apoyo y gestión administrativa a los alumnos y clientes del centro académico. Entre todas las posibles funcionalidades implementables, estas suelen tener unas en común: calendarios, reservas, pagos, autenticación de usuario, anuncios de la escuela y algunas incluso ofrecen cursos online de pago.

Varios ejemplos de las aplicaciones que se mencionan son: XtremeFunctional³, Freeletics⁴, Atalanta⁵, Activa Club⁶, etc. Todas ellas son de gimnasios, siendo ciertamente similar puesto que son los clientes quien acceden y pueden observar ciertos detalles que podrá tener nuestra aplicación, pero sin llegar a cubrir las completas necesidades de nuestro cliente.

Pueden ser un posible punto de inspiración para ayudar a construir la App conjunto a la metodología que se va a emplear. Sabiendo ya que errores hay en el mercado, se necesita mejorarlos para que nuestro producto final no cometa los fallos que ya conocemos.

Concretamente, una de las aplicaciones parecidas a la que se desea desarrollar, subida al *play store* es Danza.es⁷. Es gratuita y de libre acceso para aquel que desee información sobre el sector: eventos, festivales, jornadas, cursos, etc. Tiene una gran información a tratar, mantiene un mismo patrón para mostrar esta información y posee un sistema de filtrado (aunque mejorable) muy eficiente. Se pueden ver fallos de redimensionamiento de imágenes, superposición de elementos y algún fallo de color.

Otra aplicación que posee un sistema similar es *Millenium Dance Complex*⁸, una app que gestiona una academia situada en los Ángeles, California. En ella nos muestra el calendario de clases del día, eventos, masterclass, un calendario personalizado, etc.

La aplicación a desarrollar es una app privada, por lo que no posee una competencia directa. Al buscar las necesidades de nuestro cliente, se intentará que ninguna otra aplicación pueda amoldarse a sus necesidades más que nuestro proyecto, aportando de esta manera un producto único que solventará los problemas que aplicaciones actuales no son capaces de hacer por sí solas.

³ <http://xtremefunctional.com/>

⁴ <https://www.freeletics.com/es>

⁵ <https://play.google.com/store/apps/details?id=com.proyecto.atalantasportclub.tgcustom&hl=es>

⁶ <https://play.google.com/store/apps/details?id=com.proyecto.activaclub.tgcustom&hl=es>

⁷ <https://play.google.com/store/apps/details?id=com.semicinternet.danza>

⁸ <https://play.google.com/store/apps/details?id=com.fitnessmobileapps.milleniumdancecomplex>

3. Análisis de Necesidades de Usuario

Se desea que nuestro producto se aproxime lo máximo a lo que necesita el cliente, por ello se necesita saber cuáles son sus necesidades reales, sus habilidades, sus conocimientos, los hábitos internos de la compañía, los problemas que desean solventar e incluso algunos problemas que ellos no sean capaces de apreciar. Al no haber un número grande de usuarios será más sencillo poder acercarnos a la necesidad final.

Disponemos de cierta información que nos ayudará a entender sus problemas. Se trata de una compañía que no dispone de un sitio fijo donde ensayar, si no que recurren de la renta de salas privadas cuando necesitan realizar ensayos. Los datos que se pretende organizar son datos que manejan en general todos los miembros de la compañía (exceptuando los administrativos que solo se encarga el director). Se conoce que los datos a organizar sobre todo son fechas organizativas y mucho contenido multimedia.

Mediante varios análisis (cuya información se puede obtener en este libro⁹) intentaremos identificar todas estas pautas con el objetivo de poder proseguir con el desarrollo de nuestra aplicación.

3.1 Investigación cualitativa

Primeramente, para estrechar el vínculo con el cliente y conocer diferentes puntos, se ha decidido tomar un tipo de investigación cualitativa, de esta manera se recolectará una serie de datos (no cuantitativos) con los que podremos describir el contexto que viven los usuarios desde una forma plasmada directamente por ellos mismos.

Buscaremos la razón de la necesidad de la aplicación, cuál es el problema real. Interactuando de una forma social y directa podremos entender el tipo de usuarios que van a utilizar el producto y poder afinar el resultado.

3.1.1 Entrevistas

Como primera toma de contacto se ha decidido realizar una entrevista con cuatro de los miembros más participativos de la compañía. Gracias a ello se podrá intentar entender los problemas de una manera más natural. Se ha elegido realizar una entrevista estructurada, guiada por una serie de preguntas con el fin de sacar ciertos puntos clave para el futuro resultado.

Para amenizar la entrevista se han alternado preguntas cerradas con preguntas abiertas, un total de seis preguntas con las cuales sacar un primer barrido de información para trabajar en un próximo método de análisis. Las preguntas realizadas son las siguientes:

1. ¿Crees que existe la necesidad de una aplicación para organizar la información de la compañía?
 - Es una forma de comenzar intentando hacer sentir que las preguntas que va a tener que contestar a lo largo de la entrevista son sencillas y de fácil respuesta.

⁹ “Gracia Bandrés, M.A., Gracia Murugarren, J., Romero San Martín, D. – (2015) TecsMedia: Metodologías de diseño centradas en usuarios”

2. ¿Cuáles crees que son los problemas actuales por no disponer de la aplicación?
 - Se buscará entender que problemas tienen, por qué necesitan la aplicación y las necesidades que puede llegar a cubrir.
3. ¿Utilizas habitualmente aplicaciones de un estilo similar? Ejemplos: Notas, galería, contactos, etc.
 - De esta forma se podrá analizar la capacidad de organización del usuario. También podremos conocer el nivel de uso que le da a su teléfono móvil.
4. Nombra las cinco aplicaciones que crees que más uses a lo largo del día y explica por qué.
 - Se atenderá a las aplicaciones de uso similar a las que utilicen para garantizar una interfaz sencilla de entender y de una comodidad mayor, puesto que sabiendo cuales son las aplicaciones que más emplean se puede entender un poco más los gustos que posee a la hora de emplear una aplicación. Si prefieren un diseño bonito, una eficiencia. En resumen, entender los gustos del usuario.
5. En caso de tener que elegir una sola funcionalidad ¿Qué es lo que le gustaría que hiciera la aplicación?
 - Una pregunta reiterativa para poder analizar cuál es el problema mayor y por qué la necesidad de la aplicación. Se buscará cercar todos los problemas que puedan decir a uno solo, para poder tomarlo como posible tronco de la aplicación.
6. ¿Es consumidor de plataformas de música o vídeo?
 - Si los usuarios son consumidores de este tipo de plataformas, dada su procedencia con el arte, se podrá intentar acomodar el tratamiento de la información con una interfaz donde se vean más cómodos y le sea más interactiva de usar.

3.1.2 Análisis de las entrevistas

De las entrevistas hemos podido rescatar el problema base y la necesidad básica de la aplicación, que se trata de organizar de una forma visual, intuitiva y sencilla, la cantidad de fechas diferentes que poseen (ensayos, eventos, bolos, competiciones) mostrarla de forma que sea sencillo de entender y de una forma intuitiva para que el usuario pueda recapitular toda esa información en el menor tiempo posible.

Gracias al lenguaje verbal y las respuestas dadas a las varias preguntas sobre el uso de diferentes aplicaciones, se ha demostrado el nivel tecnológico que posee el usuario final. Un nivel elevado de uso frecuente de dispositivos de teléfono, familiarizados a entornos interactivos y visuales, de uso cotidiano y no aplicaciones complicadas por comandos. Esto se pudo constatar gracias a las aplicaciones de más uso nombradas: *YouTube*, *WhatsApp*, *Instagram*, *Gmail*, *Google Chrome*, *Facebook* y *Spotify*.

Otro problema recurrente es el almacenamiento de información de los datos que poseen, como fotografías, vídeos, música y diversa información del sector. Estos datos son consumidos con diferentes utilidades, siendo empleados desde uso individual como recordatorio, hasta usarlo de forma comercial para conseguir nuevas demandas para la compañía, lo que se hace necesario para el usuario tenerlo organizado de tal forma que pueda lograr acceder a esta información sin la necesidad de perder tiempo en buscarla en un baúl donde almacenen todo.

Y una última necesidad por parte de los administradores es la jerarquía en dos roles, líder y miembro, donde el líder tendrá unos pocos privilegios de más, entre ellos, la necesidad de poseer una estructura organizativa para llevar un seguimiento de asistencia de los integrantes a los diferentes eventos de la asociación.

3.1.3 Cuestionarios

Se ha decidido usar, otro método de análisis. Al emplear varios métodos se logrará entender en mayor medida las necesidades que buscamos, empleando un método diferente para suplir los posibles inconvenientes que una entrevista nos pueda ofrecer. Por ello, con el fin ahora de analizar todos los futuros usuarios de la aplicación se ha decidido utilizar del uso de cuestionarios con preguntas enfocadas.

Los cuestionarios facilitan la recopilación de información sin necesidad de dar explicaciones al usuario. Al ser un método basado en preguntas, evita la dispersión de la información y de esta forma el usuario posee el tiempo que desee para realizar el cuestionario. No se le posee a una presión de una entrevista directa ofreciendo una comodidad que tal vez una entrevista no pueda brindar.

Gracias al cuestionario obtenemos una información impersonal que utilizaremos más adelante para generar una persona genérica que logre resumir a los miembros de la compañía.

El cuestionario ha sido realizado por cada uno de los integrantes, lo que se ha logrado una participación del 100%.

El cuestionario realizado en cuestión está formado por 14 preguntas de respuesta única y obligatoria, como también cortas y cerradas. Las preguntas han sido desarrolladas teniendo en cuenta diversos focos: habilidades tecnológicas, consumo de productos online, uso del dispositivo y organización individual y grupal dentro de la compañía.

En cuanto al formato se ha decidido utilizar la plataforma desarrollada por google “*google forms*”¹⁰, la cual permite diseñar de manera digital cuestionarios y formularios de manera sencilla y simple. La estructura interna del formulario se han ordenado las preguntas según los focos establecidos por el orden mencionado anteriormente.

Al ser una plataforma digital es de fácil acceso para el objetivo de la encuesta, por lo que se puede acceder fácilmente por el siguiente enlace (<https://goo.gl/forms/SEqSsFIpUtCp8PLE2>), donde se puede comprobar la cantidad de preguntas, con las que se obtendrá la información, de manera imparcial para conseguir encontrar la necesidad real del usuario para el diseño posterior de la aplicación y no sus preferencias. Las preguntas mostradas han sido las siguientes:

- ¿Cuántas veces a la semana utiliza internet?
 - a. Nunca
 - b. Ocasionalmente
 - c. Diariamente
- ¿Empleas aplicaciones de consumo de música o vídeo?
 - a. Si
 - b. No
- ¿Utilizas alguna red social?
 - a. Si
 - b. No
- En caso de usarlas ¿Cuál/es?

¹⁰ <https://www.google.es/intl/es/forms/about/>

- ¿Has consumido alguna vez algún tipo de tutorial *on-line*?
 - a. Si
 - b. No
- ¿Sueles aprenderte coreografías que veas en otras aplicaciones?
 - a. Si
 - b. No
- ¿Grabas los ensayos?
 - a. Si
 - b. No
- En caso de hacerlo ¿Consumes el material grabado?
 - a. Si
 - b. No
- ¿Utilizas la aplicación “Notas” o similar de tu dispositivo móvil?
 - a. Si
 - b. No
- ¿Dónde sueles tener apuntadas las fechas importantes de la compañía?
- ¿Se suele tener un seguimiento de asistencia y participación en la compañía?
 - a. Si
 - b. No
- ¿Suele compartir información del sector con sus compañeros? Noticias, videos, etc.
 - a. Si
 - b. No
- ¿La compañía dispone de sus propios montajes musicales?
 - a. Si
 - b. No
- En caso de tenerlos ¿Dónde los sueles tener guardados?
- ¿Tienes muchas fotos de la compañía? Actuaciones, eventos, reuniones, etc.
 - a. Si
 - b. No
- ¿Utilizas alguna plataforma tecnológica para compartir datos con tus compañeros?
 - a. Si
 - b. No
- En caso de utilizar alguna ¿Cuál/es empleas y de qué forma?

3.1.4 Análisis de los cuestionarios

Con los cuestionarios, se ha logrado constatar las necesidades sacadas de las entrevistas, reafirmando aquello que se exponía e incluso sacando datos relevantes para otras necesidades que plantean la mayor parte de los miembros de la organización. La mayor parte de ellos emplea el uso de redes sociales (*Instagram* y *Facebook*) de forma activa. Publican fotos y vídeos de la compañía en sus redes por lo que se prevé la necesidad de una funcionalidad que le de la opción de hacerlo directamente desde la futura App.

Se ha constatado el nivel tecnológico medio de los usuarios, estableciendo que poseen un nivel alto. Emplean el teléfono móvil a diario y emplean aplicaciones de un entendimiento medio-difícil. Usan plataformas de tráfico de datos como *Drive* o *Dropbox*. Suelen hacer uso del calendario, por lo que plasmar el problema base en forma de calendario facilitará y ayudará a retener esta información de una forma eficiente.

Una nueva necesidad es un apartado donde puedan almacenar todo el contenido grabado en los ensayos para un posterior uso. Organizarlo de forma que puedan acceder al material con un solo vistazo y no tengan que ir vídeo a vídeo hasta encontrar el deseado. Separar este contenido, para ser fácilmente localizable ya que casi todos hacen uso de él.

3.1.5 Conclusión de la investigación cualitativa.

En resumen, se ha encontrado la necesidad de un calendario intuitivo que diferencie entre los diferentes tipos de celebraciones (ensayos, reuniones, eventos, bolos), una sección donde organice el contenido multimedia de la organización (tutoriales, videos, fotografías de eventos), una jerarquía de 2 roles (administrador/usuario) y la posibilidad de compartir el contenido por redes sociales.

3.2 Persona Primaria y Escenarios

Se dispone de la información sobre los componentes de la compañía, su edad, su género, su empleo, gracias a los cuestionarios, su nivel tecnológico y otros datos de interés que pueden ser útiles para el futuro desarrollo de la aplicación.

3.2.1 Persona Primaria

Continuando empleando técnicas del desarrollo centrado en el usuario, se procede a crear una persona primaria con el fin de reducir toda la información obtenida en los análisis de los cuestionarios y entrevistas realizadas. Esta persona será un personaje ficticio

Esta persona primaria, será un resumen de todos los datos, cuyos objetivos nos ofrecerá poseer una visión más clara de los problemas, necesidades y objetivos que plantean solventar con el uso de esta aplicación en su organización interna. Será una herramienta que acompañará, prácticamente en casi todas las fases del desarrollo.

Nombre: Eva

Apellidos: Dasi Sandiego

Edad: 25

Estado Civil: Soltera

Sexo: Mujer

Datos:

- Es integrante de Quakes Company
- Le gusta tener apuntados los datos importantes en su teléfono
- Usa diariamente redes sociales (Instagram, Facebook)
- Escucha música desde su terminal
- No le gusta faltar a los ensayos

- Le encanta el contenido multimedia, y lo tiene guardado en diferentes aplicaciones, como Dropbox, Drive, la galería de su teléfono, entre otros.
- Le gusta aprenderse coreografías de tutoriales on-line
- Graba los ensayos por si se le olvidara alguna coreografía en concreto

Tecnologías

- Samsung s7 edge

Biografía:

- Trabaja como profesora de Danza Infantil
- Vive en Valencia

Objetivos:

- Tener un orden y claras las fechas importantes de Quakes.
- Saberse con detalle todas las coreografías de la compañía.
- Compartir en sus redes sociales las vivencias que viva dentro de Quakes.
- Tener todo el contenido multimedia organizado en su teléfono.

3.2.2 Escenarios

Ya tenemos la persona primaria donde se pondrá el foco a lo largo del diseño. Nos ofrecerá una visión más clara de los usuarios finales del producto, observando sus necesidades, sus objetivos y sus problemas. Ahora, una vez obtenida la persona, se procederá a crear los escenarios.

Los escenarios son otra herramienta que nos ayudará a contextualizar a la persona creada. Esto ofrecerá una visión mucho más clara de cómo el usuario solventará el problema haciendo uso del producto a desarrollar.

Teniendo en cuenta que nos basamos en la metodología DCU, los escenarios serán breves relatos donde buscaremos entender y empatizar con el usuario para realizar diferentes hipótesis sobre las diferentes situaciones a las que se deberá enfrentar el usuario final. Dando un contexto y un uso más real.

En este caso se describe un diseño concreto desde el punto de vista de nuestra persona. A lo largo del diseño de la aplicación este escenario irá evolucionando. Construida “Eva” como persona primaria se ha procedido a crear los escenarios correspondientes de nuestra futura aplicación. Se he decidido crear directamente los escenarios específicos de cada uno de los objetivos de nuestra persona primaria.:

Escenario 1: Consulta de ensayos

Eva está algo nerviosa porque no sabe si hoy hay ensayo. Eva abre la aplicación de la compañía. En la pantalla principal puede ver un calendario que diferencia con diferentes colores las fechas y tipos de eventos que hay para los siguientes días. Eva comprueba de un simple vistazo el día de hoy que está diferenciado de otro color con respecto a los demás. Puede observar que hay un evento y definitivamente que hoy hay un ensayo. Comprueba la hora y lugar y bloquea el dispositivo.

Escenario 2: Visualizar un tutorial

Eva está preocupada porque no se sabe la coreografía del último ensayo. Eva abre la aplicación de la compañía, presiona la opción de “Tutoriales”. En la pantalla se muestra un listado con todo el contenido multimedia que visualizar organizado por etiquetas donde se muestra un título resumen del vídeo. Deslizándose la pantalla busca el vídeo que desea, con un *click* lo selecciona. Ahora ya posee el vídeo en una nueva ventana para poder visualizarlo y aprenderse la coreografía fácilmente.

Escenario 3: Publicar imagen del último evento

Eva está feliz por la competición que hizo la semana pasada. Eva quiere compartir las imágenes con todos sus amigos. Eva tiene que buscar las imágenes. Abre la aplicación de la compañía. Selecciona la opción de “galería”. Busca haciendo *scroll* el álbum concreto con la fecha y nombre de la competición. Selecciona la imagen que desea compartir. La imagen se expande mostrando tres nuevos botones, uno de ellos, con el icono de Instagram. Pulsa el botón “compartir en Instagram”. La aplicación redirige a la aplicación dicha donde después de rellenar el formulario que proporciona la otra aplicación y publicar la fotografía puede volver a la aplicación. Eva bloquea el terminal.

Escenario 4: Organización del contenido multimedia

Eva tan solo desea comprobar un par de canciones y algunas imágenes de la compañía. Accede a la aplicación y busca en la pestaña “Galería”, desliza por la pantalla observando los diferentes álbumes. Comprueba que no hay ninguno nuevo. Ahora accede a la pestaña “Música”, desliza una lista de canciones para comprobar que está todo el contenido necesario para la compañía, pulsa una de ellas para comprobar si es la canción que corresponde. Una vez comprobado, pausa la canción y bloquea el terminal.

4. Diseño y Evaluación

Como se ha mencionado anteriormente, el proceso de diseño será un proceso iterativo, donde analizadas las necesidades de nuestros usuarios, mediante técnicas de testeo, se procederán a realizar diferentes tipos de prototipos a niveles diferentes. Una vez creado el prototipo, se testeará para observar nuevas necesidades o inclusive eliminar aquellas que no sean relevantes. Una vez llegado a un prototipo claro se procederá a su consiguiente implementación con el framework remarcado.

Ahora que ya se posee una persona primaria, así como los diferentes escenarios a plantear, se dispone a crear un prototipo con la finalidad¹¹ de explorar los aspectos interactivos del sistema. A lo largo de nuestro prototipo se hará partícipe al usuario con el fin de poder desarrollar y evaluar el producto conforme se va desarrollando.

4.1 Prototipo de Baja fidelidad

En este caso, para realizar una primera idea del prototipo móvil, se van a realizar con unas dimensiones concretas. Primeramente, para tener un buen recorrido a lo largo del diseño, se realizará un prototipo off-line de baja fidelidad, fijo con una evolución rápida. La finalidad de este prototipo será poder centrarnos en la interactividad de nuestro producto y encontrar fácilmente los errores. Para realizar este prototipo se ha utilizado la técnica conocida como *Sketches*.

Gracias a los *Sketches* (podemos encontrar algo de información en este artículo¹²) podemos tener una idea genérica e inicial de cómo se comportará nuestra aplicación (su contenido, las interacciones, etc.) además podemos mostrar a nuestro usuario la idea básica de lo que sería el producto final para que pueda estar participe en nuestro diseño.

Este prototipo podrá descubrir si hemos solventado de una forma general y muy abierta los problemas razonados en el análisis. Son muy rápidos de realizar y de cambiar, por lo que, errores intuitivos a primera vista serán mucho más sencillos de solventar, el usuario se sentirá parte del proceso, no se invertirá tiempo en codificar y no estarán restringidos por normas tecnológicas de diferentes plataformas o lenguajes derivados.

Pero se tendrá en cuenta también que es un prototipo limitado, que no solventará todos los posibles casos de uso y que no se verá visualmente como el producto final, por lo que solo ayudará a una primera idea.

4.1.1 Diseño del prototipo

Para este prototipo se han tenido en cuenta las necesidades básicas de la aplicación y se ha tenido en cuenta una estructura común y básica en la mayoría de aplicaciones. No se han tenido demasiado en cuenta los principios de usabilidad, ya que como se ha aclarado, es para poseer una idea primaria de cómo funcionará la aplicación.

¹¹ Y. Hassan, F.J. Martín, G. Iazza (2004): Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información.

https://www.upf.edu/hipertextnet/numero-2/disenio_web.html

¹² <http://mpiua.invid.udl.cat/bocetos-sketching/>

Desarrollo Centrado en el Usuario para la construcción de una aplicación móvil de gestión de una compañía de danza (Quakes)

Se ha decidido emplear un software externo llamado “Balsamiq”¹³. Una herramienta completamente gratuita que nos permite poder realizar Sketches como si fueran dibujos realizados con lápiz y papel, mostrando al usuario que se trata tan solo de un proceso y está muy lejos a ser la aplicación real.

¹³ <https://balsamiq.cloud/>

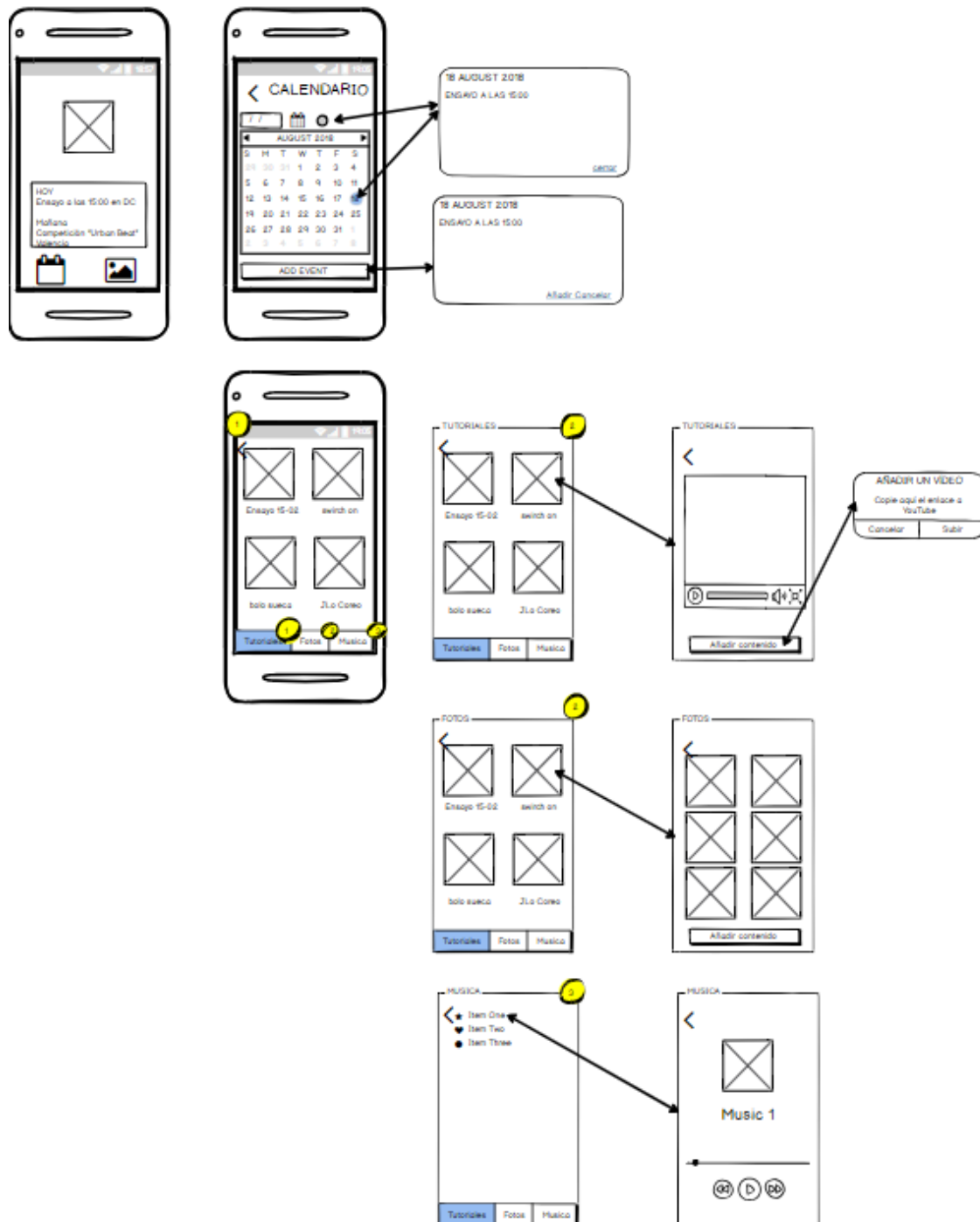


IMAGEN 1

En la figura plasmada (Imagen 1) podemos ver un primer prototipo base de baja fidelidad. Se puede ver ya cual será la estructura básica de la aplicación y en cierta manera se puede observar algo de funcionalidad y navegación. Están cumplimentados todos los objetivos remarcados, pero faltaría añadir el servicio/objetivo pedido para el rol de administrador. Para ello se ha hecho otro prototipo, que sería como visualizaría el administrador la aplicación, siendo todo igual, salvo un añadido más. Por lo que solo se adjuntará la parte nueva (Imagen 2).

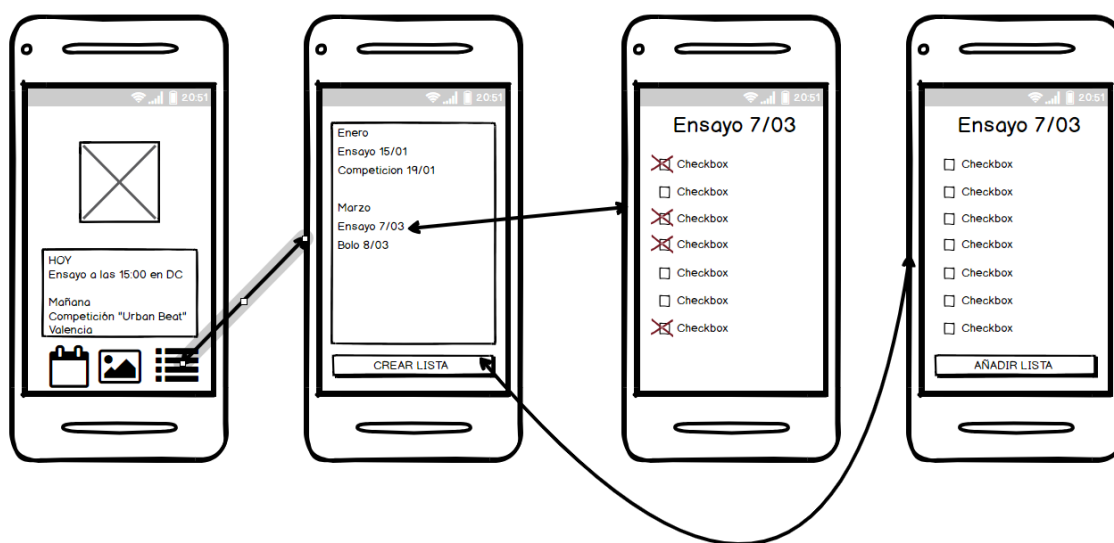


IMAGEN 2

En esta parte estará prototipada la necesidad de un servicio que ofrezca al o a los administradores la posibilidad de tener un seguimiento de asistencia en los diferentes eventos, mostrados por una tabla ordenada por orden cronológico y separada por meses, donde da la opción de crear una nueva lista con el título que más desee a su gusto organizativo.

Ya finalizado el primer prototipo de baja fidelidad, se ha procedido a evaluarlo para observar que cambios se deben realizar de forma inmediata, que necesidades se han quedado por suplir y sobre todo escuchar la opinión del usuario final real, para marcar unas pautas opcionales en el desarrollo del prototipado.

4.1.2 Evaluación del prototipo

Se trata de un primer prototipo de bajo nivel, por lo que no se ha realizado una evaluación exhaustiva del producto, ya que eso sería tan solo cargar de trabajo el desarrollo de la aplicación y dar una carga mayor a la necesaria en una etapa tan temprana del desarrollo.

Para agilizar el proceso, reducir y poseer una mayor eficiencia del tiempo, se han realizado dos formas diferentes de evaluar el sketch para sacar los primeros fallos o críticas del producto. Se han decidido hacer dos técnicas de evaluación: un informe de evaluación por inspección, donde con varias tareas se analizará la facilidad de aprendizaje y la comprensión y a la vez se ha pedido una evaluación en voz alta de cómo se sentía y que pensaba el usuario al emplear el prototipo.

Informe de Evaluación por Inspección: Recorrido Cognitivo¹⁴

Se ha preparado 6 tareas de diferente complejidad para evaluar como el usuario reacciona ante una posible estructura de la aplicación, tanto la navegación, la interacción, observar si hay elementos inútiles, la fatiga y enfado del usuario ante el error, la respuesta de este ante nuestro diseño, etc. Al tratarse de un diseño bastante simple y sin demasiada complejidad, se han tomado

¹⁴ A. Floriá Cortes (2000): Recopilación de métodos de usabilidad SIDAR Di ibl bilidad. SIDAR. <http://www.sidar.org/recur/desdi/traduc/es/visitable/inspection.htm#Cogn>

una medida bastante reducida, ya que las tareas no deberían durar demasiado tiempo debido al buen tiempo de respuesta del prototipo, así como la simplicidad del prototipo

Se le ha adjuntado los sketches de las imágenes (imagen 1 e imagen 2) en papel para que el usuario pueda ir mostrando con su propio dedo lo que haría en cada tarea.

Se han propuesto un total de cinco tareas para que logre navegar la totalidad de las funciones del prototipo:

Tarea 1 - Observar si hay un evento en el día del cumpleaños de la persona que lo vaya a usar.

Tarea 2 - Reproducir un tutorial cualquiera de algún ensayo antiguo

Tarea 3 – Añadir una imagen a cualquiera de los álbumes creados

Tarea 4 – Reproducir la última canción de la lista.

Tarea 5 – Crear y pasar una nueva lista

MEDIR	Excelente	Aceptable	Inaceptable
tiempo de la tarea	< 0.5 min	0.5-1 min	+1 min
número de errores	0	1-3	+3
Cantidad de preguntas	0	1-2	+2

Informe de Evaluación por Test: Pensado en Voz alta ¹⁵

Se va a realizar a su vez un test de pensamiento en voz alta, donde se hará partícipe al usuario que mientras realiza la tarea vaya comentando sus pensamientos tales como dudas sobre el diseño (existencias de botones o su inexistencia, detalles sobre los colores o imágenes, etc.). Se irá apuntando sus respuestas en un cuaderno para luego contrastar tanto como los resultados de las tareas y con el cuestionario post-test que tendrá que realizar después. Si algo coincide se corregirá y en el caso de que algo surja repetidas veces se valorará el modificarlo.

Resumen de los informes

Ya aclaradas las dos técnicas que vamos a emplear para evaluar nuestro prototipo, se ha utilizado a tres diferentes sujetos, futuros usuarios de nuestra aplicación (uno de ellos poseerá el rol de administrador), lo más similares a la persona primaria que se ha creado para evaluar nuestro producto y poder tener una evaluación positiva o negativa. A continuación, se mostrará el resultado de cada uno de los sujetos. En el caso de que se tenga que hacer cambios o añadir secciones bastante complejas, o con un gran peso, se volvería a realizar los test de nuevo para observar la mejora de nuestros sujetos.

- Sujeto 1 (Administrador):
 - Tarea 1: Excelente, Excelente, Excelente
 - Tarea 2: Excelente, Excelente, Aceptable
 - Tarea 3: Excelente, Excelente, Aceptable

¹⁵ <http://www.sidar.org/recur/desdi/traduc/es/visitable/testing.htm>

Desarrollo Centrado en el Usuario para la construcción de una aplicación móvil de gestión de una compañía de danza (Quakes)

- Tarea 4: Excelente, Aceptable, Aceptable
- Tarea 5: Excelente, Aceptable, Aceptable
- Sujeto 2:
 - Tarea 1: Excelente, Excelente, Excelente
 - Tarea 2: Aceptable, Aceptable, Aceptable
 - Tarea 3: Excelente, Excelente, Aceptable
 - Tarea 4: Excelente, Aceptable, Aceptable
- Sujeto 3:
 - Tarea 1: Excelente, Excelente, Excelente
 - Tarea 2: Excelente, Excelente, Aceptable
 - Tarea 3: Excelente, Excelente, Aceptable
 - Tarea 4: Excelente, Aceptable, Aceptable

En cuanto a los comentarios en voz alta han sido bastante positivos y de gran ayuda. Se han tomado en cuenta solo las opiniones recurrentes a lo largo de la realización de las tareas las cuales servirán para solventar varios fallos:

- Se hecha en falta la opción de compartir fotos
- Los botones de la página de inicio dan margen a duda
- El cuadro de los anuncios de la pantalla principal es demasiado pequeño
- Preguntan si existe la opción de poder reproducir la música sin necesidad de ir a la siguiente pantalla por si tienen que ir buscando
- No saben muy bien como volver a la pantalla principal
- ¿Qué y cómo gestionan los anuncios de la pantalla principal?
- No pueden abrir una imagen en concreto, han de ir viéndolas todas por encima.
- Preguntan si está la posibilidad de añadir tutoriales

4.1.3 Reforma del prototipo

El objetivo principal del primer prototipo era poder crear una primera y simple versión donde suplir los objetivos y necesidades de nuestros usuarios, algo que se ha realizado de forma exitosa salvo varios detalles. Gracias a los test se puede observar que la navegación es sencilla e intuitiva, algo que mejora la eficiencia de uso de nuestra aplicación. Por lo que se mantendrá en parte esa base.

Se ha podido comprobar que la navegación por iconos y el agrupar tantos servicios en el botón de galería daba margen a confusión e inclusive a un error, un error de fácil aprendizaje, pero no garantiza una buena experiencia de usuario, por ello se ha decidido realizar la técnica conocida como “*Card Sorting*” con varios de los integrantes para poder observar cómo les gustaría tener estructurada la información de la aplicación

Card Sorting¹⁶

Se han tomado a cinco futuros usuarios de la aplicación y se les ha reunido en una sala. A todos ellos se les ha explicado exhaustivamente en que funciona el método y con qué objetivo se está

¹⁶ Hassan Montero, Y.; Martín Fernández, F.J.; Hassan Montero, D.; Martín Rodríguez, O. Arquitectura de la Información en los entornos virtuales de aprendizaje: Aplicación de la técnica de Card Sorting y análisis cuantitativo de los resultados.

realizando. Se desea saber cómo es la mejor manera de estructurar el contenido y la funcionalidad, para poder buscar la mejor estructura posible.

Se han repartido tarjetas con las funcionalidades e información, todas ellas barajadas y repartidas a los cinco usuarios. En estas cartas se muestran lo siguiente:

- Imágenes
- Vídeos
- Música
- Eventos
- Listas de asistencia
- Reproductor
- Añadir eventos
- Añadir imagen

Se ha decidido emplear el método abierto, donde debían agrupar el contenido en forma de pilas y luego finalizar añadiendo una etiqueta que resumiera la pila de tarjetas. Se ha tenido una colaboración activa y de la cual se ha podido sacar una información interesante para el desarrollo de la aplicación.

Gracias a la técnica descrita se ha decidido tomar en cuenta el sistema mostrado para crear en vez de botones en la página inicial, una barra de navegación en la parte posterior con las etiquetas diferenciadas en cinco que han mostrado los usuarios: Calendario, Galería, Anuncios, Música y Vídeos.

También se ha tomado nota del deseo de volver a la pantalla inicial desde cualquier punto de la aplicación, algo que se implementará en un prototipo de mayor nivel.

Gracias a los comentarios en voz alta proporcionados por los sujetos, se ha detectado la falta de varios objetivos en el prototipo. Uno de ellos, la necesidad de compartir directamente una imagen de la galería a alguna de las redes sociales más usadas por los miembros. El otro, la posibilidad de ver una imagen en concreto y no tan solo el álbum en forma de mosaico.

Otro aspecto a mejorar es la propia pantalla principal, reduciendo el espacio del logotipo y ampliando el contenedor de los anuncios para dar mayor énfasis a este apartado, siendo casi el objetivo primordial de la aplicación.

En el apartado de música, se ha añadido la funcionalidad de poder escuchar en un pequeño reproductor la canción seleccionada o también acceder a esa canción directamente. En un futuro, en prototipos de mayor nivel se estudiará como diferenciar estas dos funciones cuando se selecciona el mismo elemento, puesto que puede llevar a error y confusión del usuario.

Por último, se ha modificado la función de añadir un vídeo, mostrando esta opción en la misma pantalla donde se muestran los diferentes tutoriales en orden. Esto ha sido debido a causa de ver el fallo cuando uno de los sujetos ha preguntado sobre dicha funcionalidad. Aprovechando dicho cambio se ha estructurado de manera más clara e intuitiva el formulario a realizar en caso de desear añadir un nuevo vídeo para facilitar el aprendizaje del usuario.

Todos estos cambios se ven resumidos en la siguiente imagen (Imagen 3):

Desarrollo Centrado en el Usuario para la construcción de una aplicación móvil de gestión de una compañía de danza (Quakes)

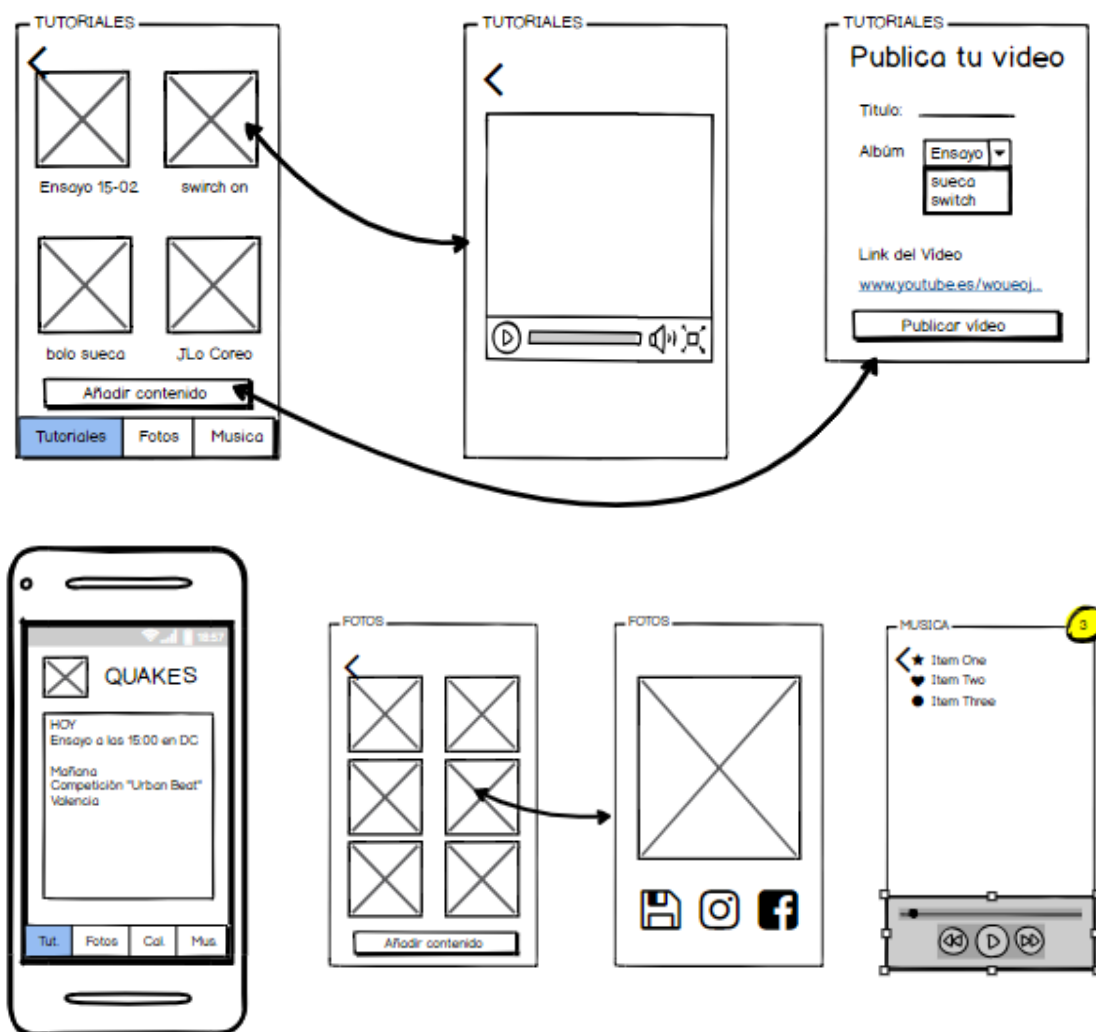


IMAGEN 3

Todo aquello que no se muestra en las figuras anteriores se respetará el diseño del primer sketch (Imagen 1).

4.1.4 Evaluación del prototipo

Esta vez, con el fin de evaluar el nuevo prototipo, se ha decidido emplear otros métodos. Ya sabemos cómo han respondido los usuarios al primer prototipo, no se quiere recurrir al mismo recurso con un mismo tipo de sketch, pero tampoco se desea extraer al usuario, por lo que con el objetivo de sacar nuevos fallos, nuevos problemas o incluso nuevas sugerencias se ha planteado realizar un método con el cuál los usuarios ya están familiarizados, los cuestionarios.

Mediante un cuestionario se han planteado cinco preguntas, casi todas de ellas de fácil y rápida respuesta, todas ellas enfocadas en encontrar el patrón que más se adecue a nuestros usuarios, intentando desvelar que posibles partes puedan dar margen a error o fatiga de uso, e inclusive intentar sonsacar algún posible objetivo o necesidad que haya quedado oculta en los procesos anteriores.

Se ha decidido realizar tan solo siete preguntas para resumir y no sobrecargar al usuario con demasiada información. Las preguntas han sido las siguientes:

1. Del 1 al 5 valora cuan de intuitiva y sencilla es la navegación de la aplicación.
2. ¿Has necesitado de una guía para entender la aplicación?
3. ¿Te ha costado encontrar algún elemento?
4. En caso de que la pregunta anterior sea "Si" diga cuál y porque
5. ¿El uso de la aplicación le ha sido gratificante
6. ¿Solventa la aplicación los problemas organizativos de la aplicación?
7. Del 1 al 5 ¿Cuál sería tu experiencia de usuario? Siendo el 1 el más bajo y el 5 el más alto.

Se ha pasado el cuestionario en formato de papel, esta vez a cinco usuarios, donde también se ha ofrecido el *sketch* y las mismas tareas que la primera prueba de evaluación para observar cual es el proceso de los usuarios que ya habían probado la aplicación y los dos nuevos usuarios.

El resultado de los cuestionarios ha sido satisfactorio. Siendo esta vez presencial se ha pedido la colaboración de los usuarios con una pequeña reseña breve y final tras el uso del *sketch* visual. Con las respuestas y el cuestionario se ha llegado a la conclusión que el diseño va en buen camino y con un proceso sencillo de entender. No es necesaria una guía de aprendizaje y la experiencia de uso es mejor a más uso.

Con los cambios los usuarios que ya habían usado la aplicación se han sentido mucho más cómodos y han experimentado que la aplicación cubre la totalidad de sus necesidades sin echar nada en falta.

En resumen y conclusión a las respuestas dadas con los cuestionarios se ha sacado la siguiente información para la próxima etapa de nuestro producto:

- La aplicación ha resultado ser intuitiva y de fácil aprendizaje
- Los elementos están más claros que en la primera versión
- Los usuarios parecen tener buena experiencia de uso
- Quedan reflejados los objetivos necesarios en la creación.

Con todo esto quedaría cerrada la parte iterativa del prototipo de baja fidelidad, teniendo una estructura la cual poder manipular. Ahora se procede a crear un prototipo de mayor nivel de detalle donde ya comience a reflejar una experiencia real e interactiva de la futura aplicación.

4.2 Prototipo de Alta fidelidad

4.2.1 Diseño del primer prototipo de alta fidelidad

El prototipo se ha realizado con la herramienta “*Justinmind*”¹⁷, la cual ofrece una amplia gama de opciones para realizar prototipos de alta fidelidad. El prototipo se ha ido creando siguiendo los objetivos y escenarios descritos en la primera etapa de análisis. También se han tenido en cuenta los resultados de los cuestionarios realizados para decidir qué ofrecer en nuestra aplicación.

Al tener como base el prototipo de baja fidelidad se ha intentado seguir la misma estructura ya que ha sido previamente analizada y testeada, mostrando ahora ya una apariencia más real e intentando acercarnos también a las facilidades que nos proporcionará futuramente el framework seleccionado.

Esta herramienta también nos ofrece ciertas restricciones a las que se ha tenido que solventar haciendo uso del *Photoshop* para intentar acercarnos lo más posible al producto final. Se ha intentado implementar ciertas funcionalidades que poseerá el usuario para luego poder testear de una manera más verídica el uso de la aplicación.

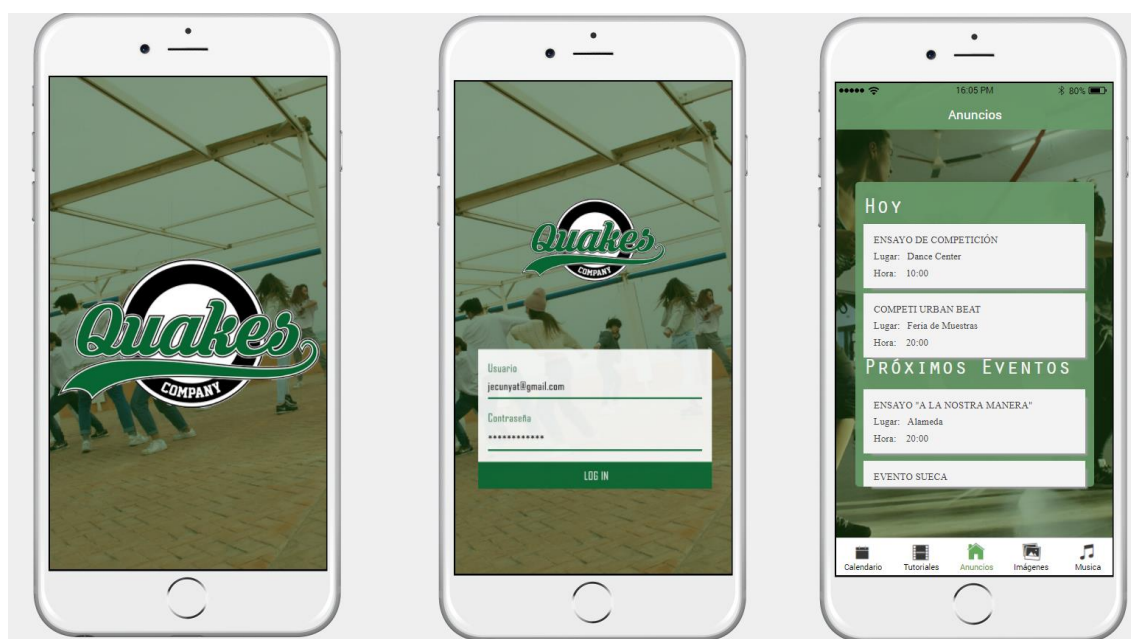


IMAGEN 4

En la imagen mostrada (Imagen 4) se muestran las tres pantallas iniciales que observará nuestro usuario de izquierda a derecha. La primera se trata de un *splash* donde tan solo se mostrarán dos imágenes: el logo de la compañía y una imagen de esta proporcionada por ellos mismos. Se ha decidido poner una capa con cierta opacidad para igualar los colores como también para darle mayor énfasis al logo y que el fondo no acapare con la atención de nuestro usuario, tan solo le dé una sensación de profesionalidad, como también buscar que sienta que la aplicación es exclusiva.

¹⁷ <https://www.justinmind.com/>

Se ha tomado en cuenta los colores del logo para tomar una paleta de colores acorde a la imagen. Siendo elegidos tres tonos diferentes de verde, acompañados con el negro y el blanco para dar un juego visual que acompaña la esencia del color del isotipo.

La segunda pantalla, tan solo será una pantalla de autenticación para llevar un control, poder dividir el usuario en dos roles diferentes. Estos usuarios serán creados por el administrador, otorgando esta funcionalidad solo a aquellos que posean dicho rol.

En esta pantalla se ha decidido mostrar el enfoque en los campos a rellenar, dándole un color diferente como una fuente personalizada de fácil lectura. Un botón de un tono diferente y de un tamaño fácilmente visible para poder acceder a la pantalla principal.

La tercera pantalla se trata de la pantalla principal, ya mostrada en el prototipo de bajo nivel. Se ha decidido retirar el título y el logotipo para no saturar al usuario con esta información. Se ha reducido la información mostrada a la mínima, dejando solo la barra de navegación donde muestra las funcionalidades básicas de nuestra aplicación y una vista donde muestra los anuncios recientes al día de hoy y de los próximos.

Con esta ventana se intenta solventar la necesidad prioritaria de la aplicación, dando al usuario la facilidad que en pocos segundos pueda ver cuáles son los eventos importantes del día de hoy e inclusive, mediante un simple *scroll*, poder ver los siguientes.

Estos eventos han sido separados como *cards*¹⁸, una opción que ofrece el servicio de *Ionic* con el cual a simple vista se puede diferenciar entre ellos. En el interior está estructurado de tal forma que de un vistazo puedes observar el título (un breve resumen del evento), el lugar donde se va a realizar y la hora acordada, de esta forma garantizamos que el usuario no posea una frustración al recibir demasiada información a primera vista.

Para hacer ver dónde está el usuario en todo momento, en la barra de navegación, tanto el icono como el título resumen estarán pintados de un color diferente al resto. Además, en la parte posterior de la pantalla, se mostrará un título de donde está actualmente navegando. De esta forma evitaremos que el usuario se sienta perdido en la aplicación (Imagen 5).



IMAGEN 5

En cambio, esta pantalla tendrá dos funcionalidades diferentes si accedemos como un usuario con el rol de “administrador”. De este modo habrá dos botones nuevos.

Uno en el margen superior derecho donde dará la opción de añadir un nuevo usuario, lo cual abrirá una nueva ventana con un breve formulario a rellenar para dar de alta nuevos usuarios en nuestra app. Se ha decidido tomar un icono ofrecido por el framework que se usará para no dificultar demasiado el proceso. Este icono muestra de forma visual y simple la funcionalidad que da el pulsarlo sin necesidad de un texto explicativo.

Por otro lado, ofrecemos un botón con el cual podemos acceder a la funcionalidad de las listas. En el prototipo de baja fidelidad se pensó en añadirlo a la barra de navegación, pero finalmente se ha decidido mostrarlo en la ventana principal, ya que no se trata de una funcionalidad tan grande como para juntarlas con las demás. En el análisis se pondrá especial atención a este punto puesto que puede ser un posible cambio a la hora de crear el siguiente prototipo.

¹⁸ <https://ionicframework.com/docs/components/#cards>

Desarrollo Centrado en el Usuario para la construcción de una aplicación móvil de gestión de una compañía de danza (Quakes)

Con el fin de mantener un orden claro, y que el usuario sienta que navega constantemente por la misma pantalla, se ha decidido poner otra imagen proporcionada en *background* con una capa de tonalidad verdosa con cierta opacidad, logrando que no resulte demasiado llamativa si no crear un entorno familiar y más profesional.

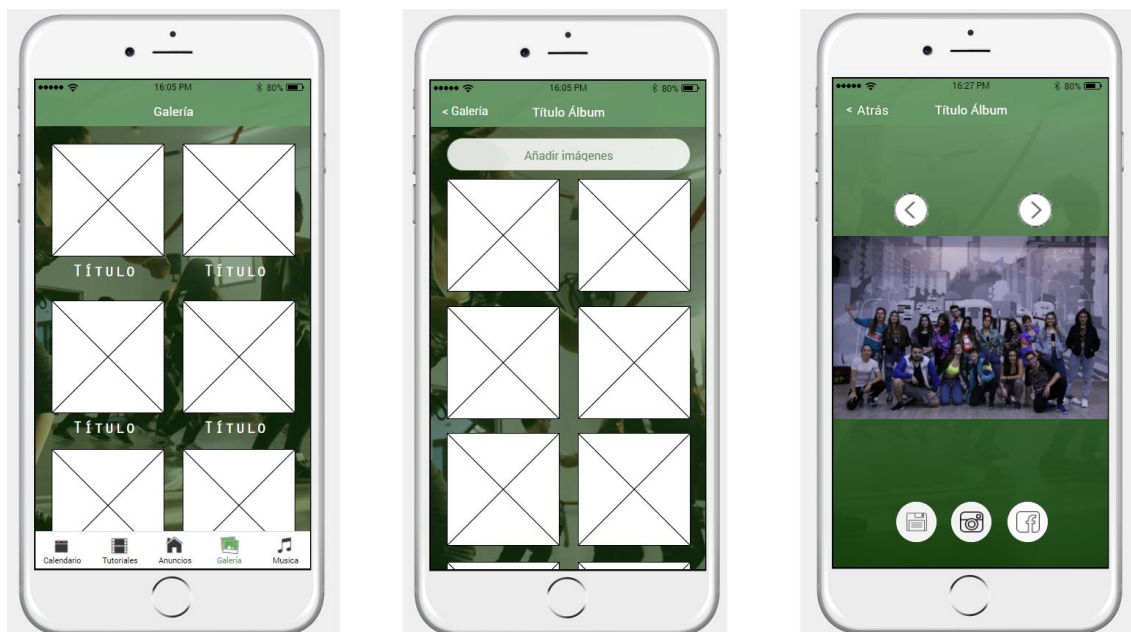


IMAGEN 6

Las tres pantallas mostradas en la imagen (Imagen 6) son las que cubrirán la necesidad de la gestión de parte de la multimedia de la organización, siendo en este caso las imágenes. Se sabe que el nivel tecnológico de nuestra persona primaria es alto, por ello se ha decidido tomar un sistema organizativo basado en álbumes y con una estructura de celdas, bastante usada en aplicaciones y soportes informáticos. Estas celdas irán resumidas por una imagen del interior del álbum, y un breve título elegido por el creador del álbum para poder, de una forma veloz poder encontrar la imagen que se desea.

Si se selecciona alguna de las celdas, la barra de navegación desaparecerá para mostrar al usuario que ya no está en el flujo principal, si no que ha salido a una rama subyacente. Dando al usuario la información para que no se sienta perdido en la aplicación.

Ahora se mostrará todas las imágenes en formato de celdas con la posibilidad de hacer *scroll* para poder ver todas ellas en miniatura. Mostrando al principio la funcionalidad de poder añadir nuevas imágenes dentro del álbum. En caso de desear cerrar y volver a la galería, en el margen superior izquierdo se da dicha funcionalidad mediante un texto descriptivo y un icono que lo acompaña.

En caso de desear ver una imagen en toda la pantalla, al seleccionarla se mostrará la tercera pantalla (la pantalla de la derecha de la Imagen 6). Se puede observar que el fondo ha cambiado por completo, para dar mayor importancia al contenido de la vista. La imagen quedará cubriendo el ancho de la pantalla. Se mostrarán dos botones para poder navegar entre las imágenes del interior del álbum seleccionado, como también tres botones con las funcionalidades de guardar, compartir en Instagram y compartir en Facebook. Siendo nuestra persona primaria usuario habitual de este tipo de plataformas, no se ha visto necesario un texto descriptivo, siendo tan solo necesario el isotipo principal de la aplicación que se desea compartir.

En todo caso, para evitar posibles errores, antes de redireccionar al usuario al lugar donde se desea exportar la imagen, saltará un aviso explicativo para evitar un posible error y obligar a una doble confirmación.

En la cabecera aparecerá el título del álbum en el cual estamos navegando y la opción de volver a él de la misma forma que se ha mostrado anteriormente, para lograr hacer entender la funcionalidad de los botones sin necesidad de ayuda.

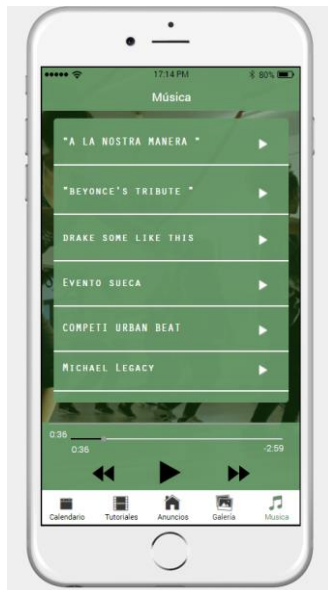


IMAGEN 7

En el apartado “Música” (Imagen 7), se mostrará una lista con todo el contenido que posee la aplicación. Se mostrará en formato tabla, con el cual se podrá navegar haciendo *scroll*. En caso de seleccionar alguna de las celdas, la canción comenzará a reproducirse mostrando un sistema de reproducción bastante simple e intuitivo, donde el usuario podrá interactuar con la canción sonante, como también ver cierta información.

En este caso solo se muestra el título del archivo a reproducir, un título resumen que habrá elegido el administrador que haya subido el contenido.

Como en toda la aplicación, en la cabecera se muestra el título de donde está actualmente el usuario como también cambia el color del icono y el texto de la barra de navegación, siempre para dar una mejor experiencia de uso.



IMAGEN 8

En el caso de seleccionar el botón “Play” de alguna de las canciones accederás a una nueva pantalla (Imagen 8). En esta pantalla ya se mostrará de una forma más visual todo el contenido de la canción. Una imagen principal, en el caso de que la tenga, el título de la canción seleccionada y de nuevo el mismo sistema de reproducción que antes, pero otorgando mayor espacio para dar cierta comodidad.

Al igual que antes, para dar la información de que no se está en el flujo principal de la aplicación se cambiará el fondo a un degradado para mostrar que se está en una rama diferente, dando la opción “Atrás” para poder ir al paso anterior (Imagen 7).

Con esto ofrecemos una opción más para poder reproducir el contenido con el deseo que el usuario tenga dos formas diferentes de reproducir la misma canción, pudiendo tan solo usar una de ellas.

Desarrollo Centrado en el Usuario para la construcción de una aplicación móvil de gestión de una compañía de danza (Quakes)



La funcionalidad del “Calendario” (Imagen 9) esta reducida a lo más simple e intuitivo posible. Posee mucha información, por ello se ha decidido reducirla lo más posible para evitar una sobrecarga al usuario. Se muestra de forma clara un calendario con diferentes puntos, acompañado de una breve leyenda para entender el porque de las marcas en el calendario.

Se ha preguntado a los administradores con cuantos tipos de marcas diferentes deseaban diferenciar los eventos, con lo que se ha respondido de dos formas: Ensayos y Eventos.

Se puede navegar por el calendario mostrando tan solo los meses. En caso de seleccionar un día donde no haya evento no aparecerá nada, en caso de seleccionar uno con una marca, se mostrará una vista donde mostrará la información del evento. En caso de haber dos en un mismo día, en forma de tarjetas se mostrarán en la misma vista pudiendo navegar con unas flechas de navegación que aparecerían.

El calendario ofrece un sistema para que el usuario no tenga que memorizar fechas concretas, tan solo con un vistazo y un simple gesto puede ver que días tiene algo importante en relación a la compañía. En caso de no entenderlo, la leyenda ofrece la información necesaria para prevenir errores.

En el caso de que se desee añadir un evento, esa función solo estará creada para aquel usuario que posea de rol administrador, donde o bien al hacer doble *tap* en una fecha en concreto, o bien al seleccionar un botón añadido al margen superior derecho, se abrirá un formulario sencillo donde resumidamente creará el evento (Imagen 10).

Se ha añadido el botón cancelar, por si se diera el caso de que no deseara crear el evento finalmente, mostrando un anuncio donde aclarase que el evento no ha sido creado. En caso contrario, en caso de cumplimentar los textos de manera correcta y crear el evento, se mostraría un anuncio donde se enseña que el evento ha sido creado con éxito para darle la sensación de que es el usuario quien posee el control de la aplicación, así como dar visibilidad al estado de la acción.

IMAGEN 10

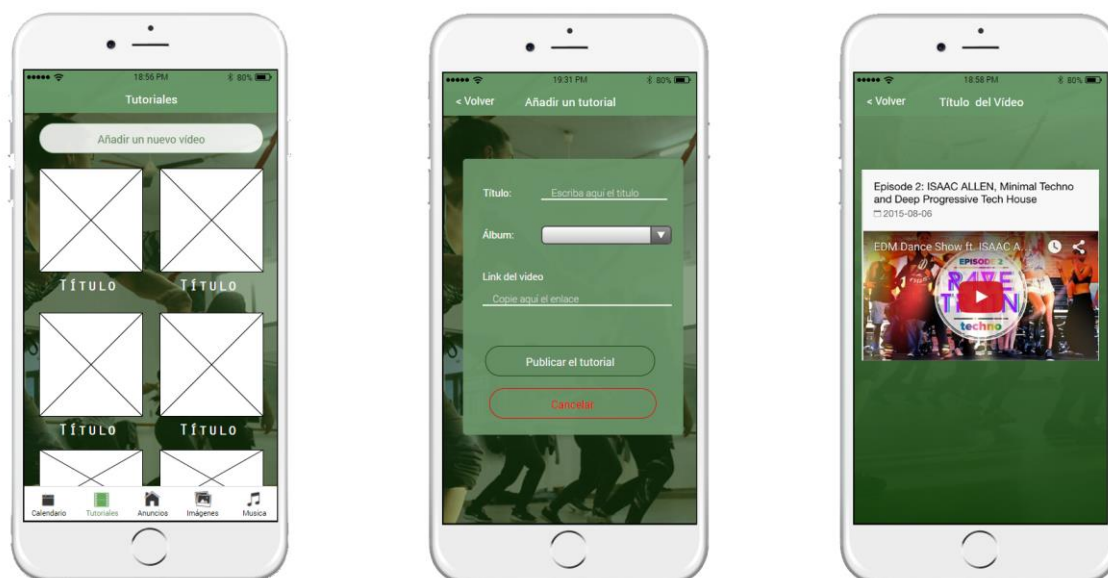


IMAGEN 11

Por último, tenemos la funcionalidad de los tutoriales. Etiquetado y diferenciado como se pudo obtener en el *card sorting*. Con el fin de que posea un aprendizaje rápido, la estructura es similar a la galería, con la diferencia que en esta funcionalidad no posee álbumes si no que directamente están los vídeos con el título resumen de estos. Todo miembro tendrá la posibilidad de poder subir vídeos mediante un breve formulario donde solo deberá poner el título del vídeo y el link de *YouTube*.

En caso de acceder a uno de los vídeos se abrirá de nuevo una ventana con un fondo diferente con el deseo de mostrar que se encuentra en un flujo diferente, se retirará la barra de navegación y se mostrará como vista principal el vídeo con el título en el formato que ofrece la herramienta *Ionic*, claro y directo.

Durante el desarrollo del prototipo se han intentado cumplir los 10 requisitos de funcionalidad de Nielsen¹⁹. Se ha intentado dar en todo momento, mediante alertas, una visibilidad clara del sistema, para que el usuario sepa que está pasando en todo momento. Con el fin de que el usuario sienta el control, es este quien posee la capacidad de añadir o eliminar contenido a su deseo, dando libertad de navegación y ocultando aquellas opciones a las que no puede acceder para no mostrar funcionalidades que no puede controlar.

Se ha seguido un mismo patrón de diseño, tanto de color, usando la misma paleta todo el tiempo. También intentar usar las mismas estructuras con vistas cuadradas, formato de celdas cuadradas y similitud entre los botones que poseen similares funcionalidades para dar una coherencia entre ellos.

Se ha intentado realizar un planteamiento donde el usuario pueda manejar las necesidades evitando cometer el máximo de los errores posibles, siendo algo intuitivo y fácil al uso para evitar posibles conflictos con el sistema en cuanto a funcionalidad y uso.

Se muestra toda la información posible, así como guías breves para que el usuario no deba memorizar, si no que tan solo con simples vistazos o gestos de *scroll* pueda ver todo el contenido sin necesidad de tener que estar reteniendo información. Esto cumple también con la flexibilidad

¹⁹ “Jakob Nielsen – (2000) USABILIDAD, DISEÑO DE SITIOS WEB”

y la eficiencia de nuestra aplicación ya que en el menor tiempo posible pueden acceder a muchas de las diversas funcionalidades del producto.

Se ha desarrollado el prototipo en base a un estilo lo más minimalista posible, siempre intentando dar una cierta exclusividad, dando un diseño único, pero a la par lo menos cargado posible para no saturar de información las ventanas, mostrando únicamente lo necesario. Se han evitado efectos y animaciones, puesto que, aunque pueden llegar a ser estéticas, pueden saturar al usuario y provocar cierto estrés.

Finalmente, con el fin de ayudar a nuestro usuario, se planteará una pequeña guía de uso para que sepan como poder usar la aplicación de primeras, siendo muy breve gracias a lo intuitiva de la aplicación.

4.2.2 Evaluación del primer prototipo de alta fidelidad

Ya se tiene el prototipo creado, por lo que se ha procedido a evaluar de varias formas diferentes. Esta vez, antes de mostrar el prototipo primeramente al usuario, se ha decidido contar con la ayuda de un experto en la materia con el fin de poder ver una primera evaluación de los fallos de nuestro sistema, ya que una persona externa, con un punto de vista objetivo y acostumbrada a tratar con este tipo de productos, podrá ver de manera sencilla y eficaz los puntos débiles de nuestra aplicación.

Informe de Evaluación por Indagación: Recorrido Cognitivo

Con la ayuda del experto, se han realizado un conjunto de tareas similares a las propuestas en la evaluación del prototipo de bajo nivel. Tareas que logren cubrir todo el campo de funcionalidades de nuestra aplicación para que, de forma objetiva, pueda indagar en los posibles y futuros errores que pueda poseer un usuario, o incluso en fallos que se haya cometido al realizar el prototipo.

Se le ha dado la persona primaria, el prototipo, las tareas a realizar y las acciones necesarias para poder llevar a cabo dichas tareas. Sin contar esta vez el tiempo para realizar las tareas, el experto ha ido realizando cada una de ellas para poder ir sacando errores corregibles para el rediseño de nuestro prototipo.

A continuación, se redactan los resultados para cada tarea realizada:

Tarea 1: Acceder a la aplicación y observar los anuncios recientes.

- El resultado ha sido satisfactorio sin necesidad de añadir, modificar o retirar ningún elemento.

Tarea 2: Reproducir una canción

- Se han detectado varios errores en el proceso. Los botones de reproducción rápida deberían de ser color blanco para no romper el esquema de paleta de colores. No hay forma de saber que sucede si pulsas al *play*. Cuando se pulsa para una reproducción rápida no se muestra el título de la canción que está sonando. Se necesita de un botón para añadir nuevas canciones, con su correspondiente formulario.

Tarea 3: Abrir una imagen y compartir con *Instagram*.

- El experto ha mostrado el fallo de que no se pueden añadir nuevos álbumes. El botón de guardar no deja claro que pasará si lo pulsas. En caso de compartir en *Instagram* no hay ninguna alerta que de constancia que ha sido publicada. No está la opción de poder eliminar una foto o un álbum.

Tarea 4: Añadir y reproducir un nuevo tutorial.

- En el formulario de “añadir tutorial” aparece álbum, cuando en esta opción no están divididos de esta forma. No está la funcionalidad de poder eliminar un tutorial. Carencia de alertas informativas en caso de existir un error al añadir un vídeo. Falta de la funcionalidad “editar” un posible tutorial.

Tarea 5: Encontrar una fecha concreta, observar si existe evento y crear un ensayo para ese día buscado.

- No existe la forma de eliminar o editar un evento ya creado. Se desconoce el año del mes que se está observando. Se debería aclarar el formato de la hora para tener un mismo estándar. Debería existir una funcionalidad donde en caso de navegar por el calendario pueda volver rápidamente al día de hoy.

Tarea 6: Crear una nueva lista de asistencia.

- Un error bastante grave puesto que existe el botón que muestra esta funcionalidad, pero no se ha desarrollado las vistas que cumplan este objetivo.

Gracias a la evaluación dada se ha podido detectar todos los fallos primarios del prototipo, por lo que antes de pasar a una evaluación de uso por parte del usuario real, primeramente, se ha decidido suplir estos fallos e intentar mejorar la aplicación dada las pautas por el experto.

También se ha añadido la funcionalidad de poder cerrar la sesión si se desea, al igual que la opción de borrar un usuario creado para el rol de administrador, para ello, en vez de un solo icono, se plasmará un menú configuración para el rol administrador donde se pondrán estas funcionalidades extras que no posee un usuario normal del producto.

Se ha decidido subir la opacidad del fondo de la aplicación para dar un mayor énfasis en la parte donde se ha de fijar el usuario y no en el fondo de la vista, con el fin de simplificar aún más nuestra aplicación.

4.2.3 Reforma del prototipo

Se han ido realizando los cambios según las tareas y los análisis propuestos por el experto en la etapa de evaluación anterior.

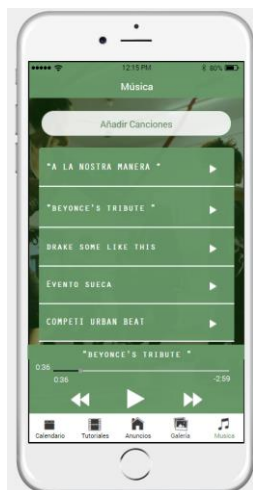


IMAGEN 12

Siguiendo con ese orden, el primer cambio realizado es en la parte de música, donde se han cambiado los botones a una tonalidad blanca (Imagen 12). Se ha añadido en la reproducción rápida el título que está sonando para quitar el posible error de confusión.

Se ha creado un botón en la parte superior para poder añadir nuevas canciones junto un pequeño formulario similar al de añadir imagen donde directamente accederá a la multimedia del teléfono donde podrá acceder a la canción que desea subir, y saltará una alerta informando del estado del sistema, donde dará visibilidad de si hay sido realizada o no y explicará el motivo.

Se ha añadido la opción al administrador de poder borrar alguna de las celdas de la tabla con un simple *scroll* de derecha a izquierda, mostrando primeramente en rojo que si desliza borrará el contenido, y en caso de hacerlo, saltará una alerta donde deberá confirmar si realmente desea eliminar el contenido, para evitar un posible fallo donde el usuario borre alguna canción por error.

A continuación, para solventar los errores vistos en la vista de galería se ha procedido a añadir un botón similar al de las demás pantallas en lo alto de la vista para poder crear un nuevo álbum. A su vez, si se pulsa se abrirá un *pop Up* donde solo habrá que especificar el título del álbum a crear. En caso de crearlo saltará un aviso como que ha sido creado y en caso de cancelarlo tan solo se cerrará el formulario. Además, en el margen superior derecho, se ha añadido el botón "Editar" donde si se pulsa, se tendrá la opción de poder eliminar cualquiera de los álbumes creados e incluso poder editar el título. Esta función se muestra para todos los usuarios, ya que puede dar el caso de que se equivoquen al crear una celda.

Se han añadido alertas a los tres botones, para dar una doble confirmación y que el usuario sepa que se va a realizar en caso de aceptar al pulsar el botón. En cualquiera de los casos, una vez realizada la acción, salta una nueva alerta explicando el estado del proceso, en caso de acierto o fallo, para darle constancia de que ha sucedido con su objetivo.

Visto el fallo del formulario, se ha decidido retirar esa parte de la vista para no dar confusión, ya que los vídeos no están separados por diferentes grupos si no que se muestran directamente, ya que, gracias a las entrevistas y el análisis se conoce que no se poseen dos mismos vídeos de un mismo material. Se ha añadido el mismo botón de editar, con la misma funcionalidad, para poder eliminar o editar el título/link de alguno de los tutoriales.

Se han añadido diferentes alertas para cuando se edite, se cree o se elimine un vídeo. Estas alertas proporcionarán la información necesaria para que el usuario sepa que ha pasado con su acción de forma directa y que posea consciencia del estado de nuestra aplicación. Uno de los principios de Nielsen.

Se ha preguntado quien va a poseer la capacidad de poder crear, editar y eliminar eventos en el calendario, por lo que se consta que solo el administrador posee dicha función. Por ello, para poder eliminar o editar algún evento concreto, cuando se selecciona, en la pantalla donde muestra

la información, en el caso de ser administrador, aparecerán dos botones añadidos: editar y eliminar (Imagen 13).



IMAGEN 13

Ambos botones al ser presionados poseen funcionalidades diferentes, pero ambos mostrarán alertas cuando se haya acabado de trazar la operación (Imagen 14). En el caso de editar, se abrirá

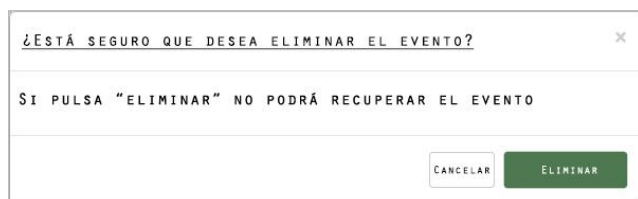


IMAGEN 14

una ventana de diálogo con la información actual del evento que se va a editar, y podrá editarse. Esta ventana mostrará la opción de guardar cambios, o bien de descartar, mostrando la respectiva alerta cuando pulse cualquiera de los dos botones.

La opción eliminar, al ser pulsada mostrará otra ventana de diálogo para hacer una doble confirmación de que, si realmente desea borrar el contenido (Imagen 15), puesto que un evento borrado ya no es recuperable a no ser que lo vuelva a crear, y para no caer en el error de borrar alguno sin deseo de hacerlo, se realiza esta doble confirmación a través de la alerta.



IMAGEN 15

Se han añadido a su vez el año correspondiente al mes que se está observando como un botón en el margen superior izquierdo para que se pueda volver al día de hoy con un simple gesto con el fin de facilitar una posible tarea que podría provocar futuramente un estrés innecesario en el uso de nuestro calendario.

Por último, pero no menos importante, se han prototipado las vistas en relación a la funcionalidad de la creación de listas de asistencia. Se han seguido los mismos estándares que a lo largo de todo el diseño del prototipo (colores, fuentes, fondo, bordes, etc.). Como también se ha tomado la estructura ya comprobada de nuestro prototipo de baja fidelidad.

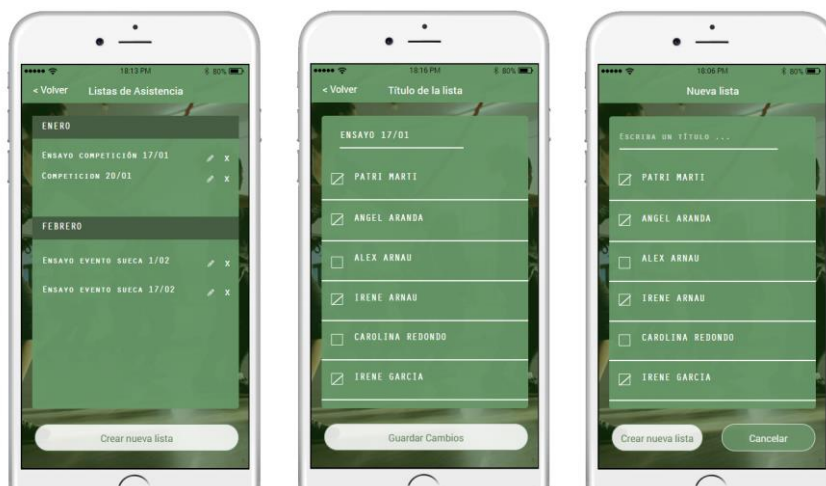


IMAGEN 16

En la imagen anterior (Imagen 16) se pueden encontrar las tres pantallas prototipadas de las listas de asistencia para el rol de administrador, accesible desde el botón mostrado en la pantalla de inicio para aquel usuario que posea dicho rol.

Siguiendo un orden de izquierda a derecha se muestran tres pantallas que cubren tres funcionalidades diferentes que son agrupadas en la misma necesidad de nuestra persona primaria. Tomando en cuenta la primera de ellas (la pantalla de la izquierda de la imagen 16) se muestra, dividiendo la información por mensualidades, las diferentes listas que han sido creadas, pudiendo, mediante un gesto, pulsar dos botones informativos que dan dos opciones diferentes: editar y borrar. En caso de poseer una cantidad superior a la que nos ofrece nuestra pantalla, se podrán acceder a todas ellas mediante un *scroll* incorporado. Esta herramienta está habilitada a lo largo de toda nuestra aplicación, por lo que el usuario ya está acostumbrado a usarla.

En la pantalla del medio, mostramos la opción de poder editar alguna lista en concreto, donde se puede cambiar tanto el título, como seleccionar o eliminar alguno de los asistentes del día marcado. Se muestra un botón al final donde poder salvar el contenido modificado, con su correspondiente alerta de confirmación para evitar futuros errores. También se muestra, en el margen superior izquierdo, la función de volver, donde se mostrará otra alerta de confirmación para que nuestro usuario posea conocimiento que se descartarán los cambios, en caso de que los haya.

Por último, la tercera pantalla (la de la derecha de la Imagen 16) cubre la necesidad de crear nuevas listas, donde por defecto guardará el día en el que se crea. Podrá elegir el título que desee. Se ha oscurecido el tono del *place holder* para que el usuario pueda entender donde ha de colocar el título. Una vez rellenada la información, puede crear o cancelar la lista. Ambas opciones con su correspondiente alerta y doble confirmación para, como se ha descrito, prevenir errores y dar visibilidad a cómo reacciona el sistema a la acción hecha.

4.2.4 Segunda evaluación del prototipo

Esta vez se ha decidido evaluar el prototipo de nuevo empleando la misma técnica empleada en el prototipo de baja fidelidad. Se busca ahora un contacto real con los futuros usuarios ya solventado los errores del desarrollo de nuestro prototipo. Se han dado las mismas tareas a realizar que se ofrecieron al experto que realizó la evaluación anterior para suplir todos los objetivos a los que puede abarcar la aplicación y no saturar al usuario con demasiadas tareas.

Para poseer más información la cual poder analizar en vistas de mejorar nuestro producto, se ha pedido que el usuario, de manera individual, comente en voz alta aquello que piense, tanto de manera positiva como de forma negativa. Aspectos personales, detalles, inquietudes, cualquier detalle que pueda ser relevante para un cambio en nuestro producto ya en la fase de desarrollo.

En caso de que el resultado de nuestra evaluación no sea de mayor envergadura, que posea pocos errores o detalles a modificar, se proseguiría ya a la programación de nuestra aplicación directamente en el *framework Ionic*.

- Tarea 1: Acceder a la aplicación y observar los anuncios recientes.
- Tarea 2: Reproducir una canción
- Tarea 3: Abrir una imagen y compartir con *Instagram*.
- Tarea 4: Añadir y reproducir un nuevo tutorial.
- Tarea 5: Encontrar una fecha concreta, observar si existe evento y crear un ensayo para ese día buscado. (crear evento solo para los administradores)
- Tarea 6: Crear una nueva lista de asistencia. (solo para los administradores)

Esta vez, se ha tenido en cuenta también que el usuario ya conoce, en parte, la estructura de la aplicación, ya que testearon el prototipo de baja fidelidad, esto causa que se haya tenido en cuenta un menor número de errores en el proceso de las tareas.

Los resultados del test para medir las prestaciones del producto han sido satisfactorios. Sin necesidad de ayuda, los cuatro usuarios (dos con el rol de administrador) han logrado resolver las tareas propuestas apenas con uno o dos fallos en total. Se puede observar que posee un sistema intuitivo y de fácil aprendizaje.

5. Implementación

Una vez realizadas las fases iterativas de evaluación de nuestro prototipo de alta fidelidad, se ha proseguido a implementar nuestra aplicación ya con un código funcional con el deseo de obtener el producto real que usarán finalmente nuestros clientes. Para ello, como se especifica al principio del documento, se ha decidido emplear el *framework* conocido como *Ionic*. Se empleará la versión más actualizada para poder acceder a las opciones nativas que ofrece, puesto que varias de ellas ayudarán a la implementación.

En este apartado se redactará la estructura seguida para programar la aplicación, explicando los puntos conflictivos y como han sido solventados gracias a los conocimientos adquiridos en el grado. A su vez, se redactarán posibles cambios de diseño que afecten a la usabilidad debido a los límites que pueda ofrecer los lenguajes que se van a emplear: *AngularTS*²⁰, *Angular2*, *HTML*, *TypeScript* y *CSS*.

Todas las referencias a llamadas nativas del *framework* o componentes utilizados lo puedes encontrar en la documentación de Ionic Native²¹ y Ionic Components²².

Al final del documento (Anexo 1) se ofrece un manual de usuario para que se pueda entender cómo funciona la aplicación que se va a describir y se pueda contemplar el apartado visual.

5.1 Arquitectura del sistema

La forma que proporciona Ionic3 para poder implementar la aplicación consiste en un sistema de páginas, módulos y *providers*. Esto obliga a que no podemos seguir un sistema de clases tradicional que nos puede proporcionar un lenguaje nativo (por ejemplo: Android).

Cada una de nuestras páginas poseerá tres archivos fundamentales:

- **.html**: Este será la vista que se mostrará al usuario. Se utiliza el sistema de etiquetado para estructurar los diferentes elementos. Se ayuda de patrones que proporciona Ionic para poder implementar y gestionar la información que se muestra en estos elementos. Está controlada por el archivo TS, donde se implantará la lógica, y la estética por el archivo .css, donde se implantará el diseño.
- **.ts**: En este archivo implantaremos la lógica de nuestra aplicación, las variables, los módulos externos, los métodos privados o públicos etc. Trata a la vista como una clase pudiendo ser invocada por otras vistas, por lo que cada una de las páginas que se va a redactar son las clases del proyecto. Se empleará el lenguaje TypeScript para poder implementar la lógica.
- **.scss**: en este archivo, mediante css, se especificará el diseño del archivo .html (color, fuente, tamaño, etc.)

Los *providers* serán aquellos objetos que sean compartidos entre varias vistas, los cuales puedan manejar ciertos métodos o datos que deseemos acceder desde varias clases y nos ayude a reducir la cantidad de código y poseer una estructura más limpia.

²⁰ <https://angular.io/guide/typescript-configuration>

²¹ <https://ionicframework.com/docs/native/>

²² <https://ionicframework.com/docs/components/>

5.2 SplashScreen y fase de Login

Se posee la imagen que se desea para la primera vista que se mostrará de una forma temporal. Se ha decidido usar este recurso por diversas razones, pero la más prioritaria es evitar una mala experiencia de uso, evitando una posible angustia por parte de nuestro cliente ya que nuestro *splash* será una imagen intermedia entre el escritorio del teléfono y la aplicación, que se mostrará mientras se carga toda la información necesaria.

Una vez mostrado el *splash* se muestra la primera pantalla, la fase de autenticación. Se ha decidido implementar la funcionalidad que el usuario pueda registrarse desde esta pantalla. Para llevar a cabo este proceso, se ha creado un *provider* “*auth-service*” el cual será el encargado de manejar los datos y poseerá en el las funciones necesarias para conectar con el *backend*, el cuál será el que dará como respuesta dos parámetros: un booleano para confirmar o no la acción, y un tipo, para diferenciar el tipo de rol de usuario, en caso de acceder a la aplicación.

Los métodos que implementa el *provider* son los siguientes:

- **login (credentials):** se le brinda como parámetros las credenciales: usuario y contraseña, los envía al servicio correspondiente y espera la respuesta. Una vez obtenida la entregará a la clase desde donde se haya llamado.
- **register ():** similar al *login* pero con unos credenciales que no estarán registrados, se almacenará en la correspondiente base de datos que se elija para poder acceder posteriormente con dichos credenciales.
- **getUserInfo ():** devuelve la información del usuario actual.
- **logout ():** cierra la sesión del usuario actual.

A su vez se ha creado un objeto *User* el cual será el que contenga la información y las variables necesarias para procesar esta información.

Una vez implementado el *provider* se han creado dos páginas por terminal con sus vistas (.html) y sus controladores (.ts): *login* y *register*. Para estas pantallas se han tenido en cuenta el diseño propuesto en el prototipo. Ahora, con el fin de mejorar nuestra usabilidad, se han añadido alertas para avisar al usuario en caso de fallo, y poder guiarle para dar una mejor experiencia de uso. Los casos cubiertos han sido:

- En caso de un email que no corresponda al formato conocido [aaa@xxx.xxx](#).
- En caso de una contraseña menor a 6 caracteres.
- Si existe fallo en autenticación comprueba y muestra en orden estos errores:
 - El email no existe
 - La contraseña no coincide con el email
 - El servidor no puede procesar esta operación, contacte con un técnico (este proceso es si hay un posible problema en el servicio que no sea debido a un mal uso por parte del usuario)

Para evitar posibles errores, los campos están programados para eliminar espacios en blanco antes y después del texto, en caso de introducirlos en otro lugar, avisará de que no está permitido.

Desarrollo Centrado en el Usuario para la construcción de una aplicación móvil de gestión de una compañía de danza (Quakes)

En caso de aceptar el acceso, redirigirá a nuestro usuario a nuestra página principal. En caso de fallo, mostrará la alerta con la información acorde al fallo y se quedará en la pantalla de acceso.

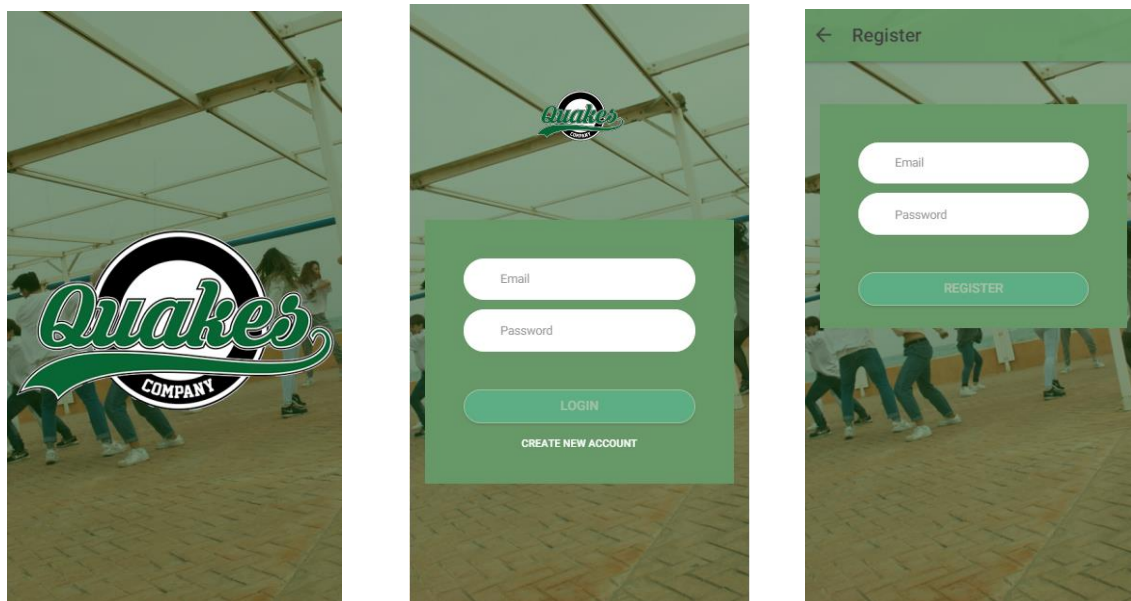


IMAGEN 17

5.3 Navegación: Creación del *TabsPage*

Se definió en la etapa de desarrollo que la estructura de nuestra aplicación estaría formada por cinco ventanas diferentes con cada una de las necesidades primarias que cubrir. La forma de navegar por estas ventanas es muy importante para que sean de fácil acceso, y no solo eso, si no que también el usuario tenga consciencia de donde se encuentra.

Para solventar este problema se ha decidido usar un navegador en la parte inferior de la pantalla, donde muestra un resumen con un icono y un breve texto descriptivo de donde conducirá la aplicación al usuario en caso de ser pulsado. Con el deseo de dar esa constancia de donde esta localizado actualmente, el botón y el texto se verán cambiados a uno de los colores definidos en nuestra paleta de colores para diferenciarse de los demás botones y mostrar al cliente en que punto de la navegación se encuentra.

Con el fin de consolidar este punto, también se ha decidido mantener la cabecera, mostrando el título de la vista que se está observando en ese momento. De esta forma, mire en la parte superior o inferior, se mostrará la información de donde se encuentra, dejando el centro de la pantalla libre para la información a mostrar.

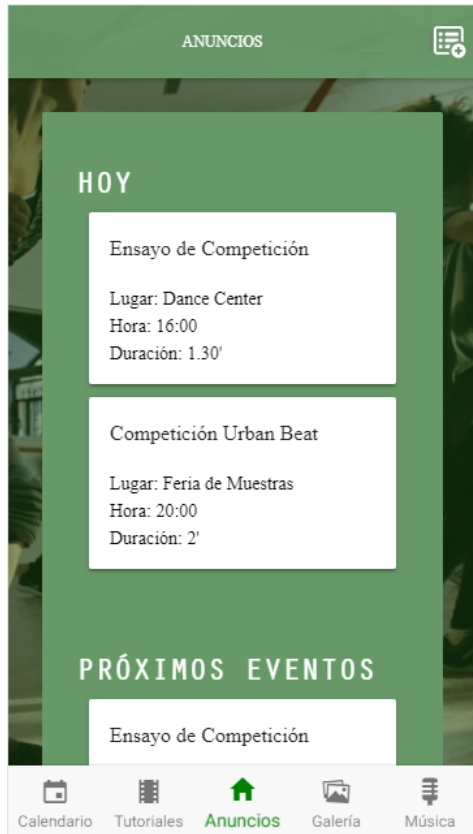
Las vistas serán cargadas, es decir llamarán a los servicios en caso de que los haya, en el momento que el usuario desee acceder a ellas, con esto, reduciremos el coste de recursos del terminal, y el tiempo para abrir una ventana será mucho menor al principio que si ha de esperar a que las cinco terminen de cargar.



IMAGEN 18

5.4 HomePage

Esta será la primera pantalla (que ya está enfocada en cubrir los objetivos de nuestro cliente) que se visualizará. Por ello tan solo muestra una breve información de la necesidad principal de recordar que eventos cercanos son los que posee en relación a la compañía.



A diferencia del prototipo, se ha dado la funcionalidad de poder registrarse, siendo que la gestión de este tipo de información la tratará el *backend*, por ello, se ha prescindido del botón que se sugirió para añadir nuevos usuarios a la plataforma, en su defecto, en esa parte se ha implementado un botón que se mostrará a aquel usuario que posea el tipo administrador.

Este botón al ser pulsado redireccionará al usuario a la ventana donde se mostrarán todas las listas creadas y almacenadas en el servidor, dando la opción de poder editarlas, eliminarlas e inclusive, crear nuevas listas.

Por otro lado, en cuanto a la información que ofrece esta pantalla, será recibida por parte del backend. Se espera recibir la información en formato “. json”, el cual será mapeado a un objeto *event*, el cual tendrá los datos necesarios (título, lugar, hora y día) para poder mostrarlo utilizando el componente de Ionic, Card.

La pantalla está dividida en dos contenedores, siendo el fondo donde se muestra la imagen fija y predeterminada en las cinco vistas y un contenedor dividido en dos secciones: eventos para el día actual y eventos para los próximos días.

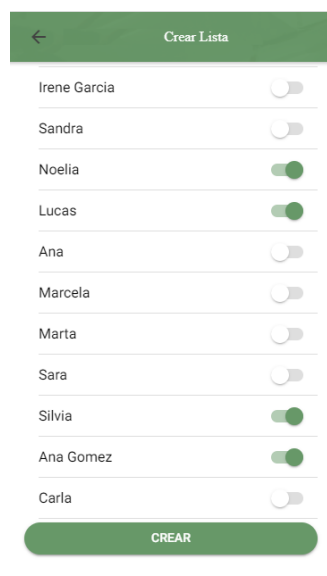
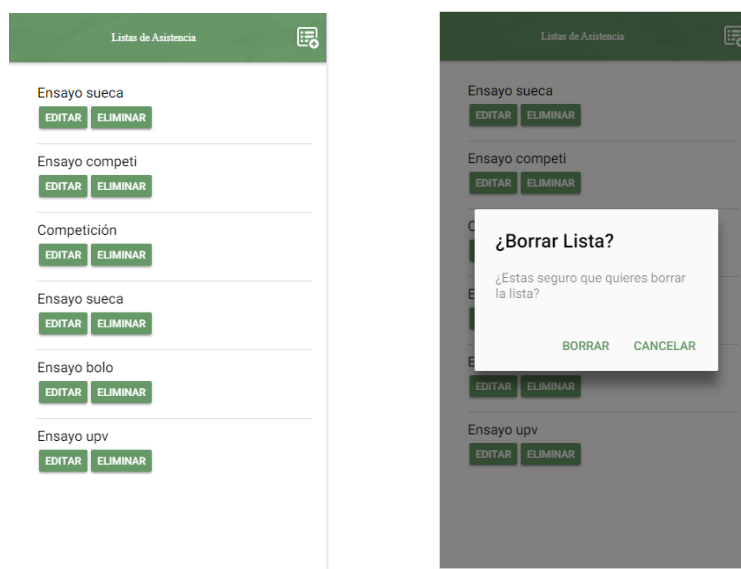
Los eventos del día actual no estarán limitados, mostrando todos y cada uno de los que existan, en cambio habrá un límite de cinco para los próximos eventos, para no saturar la pantalla de cartas.

En la pantalla, tan solo el contenedor que contiene toda la información, es el que posee la funcionalidad activa de poder hacer *scroll*, siendo que se oculta tras el *header* y tras la barra de navegación. Con esto, evitamos que el título o la navegabilidad de nuestra app desaparezca momentáneamente, por lo que, con menos gestos, podrá hacer la misma funcionalidad.

En caso de que se diera el caso que no existiera ningún tipo de evento, se mostraría un mensaje para informar del estado del sistema y no diera margen a pensar que es un problema del teléfono o de la aplicación, si no que no hay contenido que mostrar.

Listas de Asistencia

Si se accede a esta funcionalidad, se muestra un listado recibido por parte del servidor de todas las listas creadas y guardadas. Estas listas pueden ser editadas y eliminadas, y toda acción conllevará una llamada al servicio que según la respuesta mostrará un mensaje emergente para dar visibilidad del estado de la acción realizada por parte del usuario.



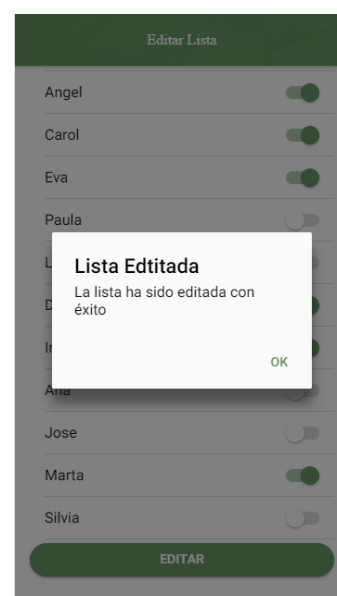
Pulsando el botón de la cabecera, se mostrará la opción de poder crear nuevas listas. Se obligará al usuario a poner un título resumen y podrá seleccionar las personas que han asistido mediante un *switch*. En caso de desear eliminar alguna celda, se ha implementado la funcionalidad para que pueda deslizarla hacia la izquierda y eliminar el ítem del listado.

Una vez creada puede validar el contenido con el botón crear o bien volver a la pantalla anterior sin guardar los cambios. Para evitar un posible fallo, saltará una alerta mostrando que sucederá en caso de que vuelva atrás sin guardar los cambios.

Pulsando el botón “Editar” de cualquiera de las listas, se mostrará al usuario una ventana muy similar, cargando el contenido de la lista para poder ser visto, y dando la opción de

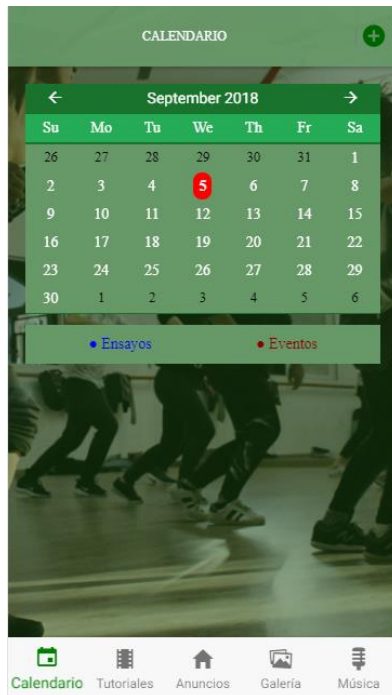
poder ser editado, tanto su título como también el check de cada uno de los miembros.

Lo que no se podrá realizar es añadir miembros que han sido eliminados cuando se creó la lista, ya que se entiende que si no fueron añadidos en su momento es porque no estuvieron, ni debían estar presentes.



5.5 Calendario

Se han tenido en cuenta los detalles propuestos en la última evaluación de nuestro prototipo para llevarlos a cabo. Por lo que se han tenido que tener en cuenta diversos casos para poder implementar el calendario.



Al abrir la ventana, se mostrará tan solo el calendario, cargando el mes actual, y pintando de un color diferente el día actual. En caso de que haya eventos creados, serán mostrados con puntos debajo de los números del día con dos colores diferentes. También se muestra una breve leyenda distinguiendo entre los dos colores para facilitar el entendimiento y evitar que tenga que recordar cual era cual.

También se cargarán el mes anterior y el posterior, para mostrar los días y cubrir por completo toda la gráfica, aunque estos días se mostrarán de un tono más oscuro para mostrar al usuario que son fechas no pertenecientes al mes que estará visualizando.

Se ha añadido la funcionalidad que, si pulsa alguno de estos números, cambiará directamente al mes de la fecha seleccionada, actualizando los tres meses (actual, anterior y posterior) y actualizando los eventos del mes.

En caso de existir un evento en el día de hoy, se mostrará una ventana en la parte inferior de la pantalla (debajo de la leyenda) donde mostrará la información del objeto evento. En caso de ser administrador, también dos botones, editar y eliminar.

Estos eventos están almacenados en el backend, se pensó en tenerlos almacenados también en memoria local, pero podría darse la situación donde se añade un evento, otro usuario accede a la aplicación y en cambio ha fallado la conexión y solo se han mostrado los eventos guardados en su sistema local, por lo que no sabría de este nuevo cambio.

Existe un botón en la parte superior derecha, donde muestra un icono para poder añadir un nuevo evento. En caso de ser pulsado, el controlador nos redireccionará a una nueva ventana donde se muestra una ventana modal con los datos necesarios para crear un evento, al igual que en la fase de autenticación y registro, en caso de añadir un campo mal se avisará con mensajes de alerta para dar imagen de donde está el fallo.

Para evitar posibles errores de formato, se han programado las celdas para que admita cierto tipo de formato. En el caso de la fecha se ha obligado a usar un selector de fecha, para que llegue en el formato que podrá tratar la API y no se tenga que transformar.

Esta ventana posee en el margen superior izquierdo un botón con el título “Cancelar” que, en caso de ser pulsado, muestra una alerta que avisa que pasará si prosigue y en caso de desearlo cancelar el evento y volver al calendario.

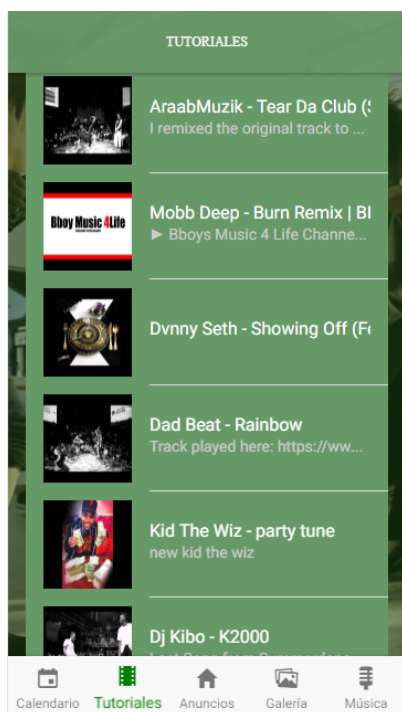
En caso de querer editar un evento, abrirá esta misma ventana modal donde el título será diferente a si se accede desde el botón “añadir”. Los datos ya estarán rellenos por el evento a editar, y una vez guardado, se envía al backend, en caso de respuesta positiva, redirecciona al calendario ya actualizado mostrando este nuevo evento.

En el caso de desear eliminar uno de los eventos, si se selecciona eliminar, para dar robustez a nuestro código, primero se muestra una doble confirmación, para evitar que se haya pulsado de una manera no intencionada el botón. Si se acepta se hará una llama al servicio correspondiente, que sea cual sea la respuesta, se mostrará un mensaje con la información para saber si el proceso ha sido completado o no.

Para poder implementar este proceso, se ha hecho uso de una librería nativa que ofrece Ionic llamada “Calendar”. Esta librería ofrece apoyo a nuestra funcionalidad ya que posee funciones para crear los eventos y mostrarlos en el calendario customizado que se ha creado.

5.6 Tutoriales

Primeramente, se pensó en mostrar los tutoriales en forma de cuadrícula, pero esto resultaba costoso y algo ineficiente. Esto era debido a que para usar este formato se debía de redimensionar cada una de las imágenes para que quedasen de forma ecuánime, y al llamar a un servicio API de YouTube. Este proceso hacía un tiempo de espera que podría llevar a generar incomodidad por parte del usuario.



Para solventar el problema se ha decidido mostrar los tutoriales en forma de listado, donde en este listado se muestra el título y una breve descripción del vídeo a mostrar.

Al igual que en la pantalla principal, se muestra el mismo *scroll* con el fin de intentar unificar todos los gestos y que el proceso de aprendizaje sea mucho menor.

En cuanto al código, se ha empleado de la librería nativa de Ionic que conecta con el servicio de Youtube. También, utilizando Google Key, se ha creado una llave API para poder acceder a la API abierta que ofrece la propia plataforma.

Para poder gestionar las llamadas, se ha tenido que configurar los archivos de configuración del proyecto, para poder llamar a la llave correspondiente y poder de esta forma realizar las peticiones necesarias.

Finalmente, para tener una mayor fluidez de los datos, se realizó de nuevo una reunión, donde el cliente indicó que poseían una cuenta en YouTube, donde en una *playlist* tenían los vídeos que querían ir subiendo. Por ello, se ha decidido

que, al entrar en la vista de tutoriales, directamente carga los vídeos de esta *playlist* en formato de lista.

Gracias a esto, si se desea borrar, editar o añadir un vídeo, se hará directamente desde la plataforma de YouTube, en la cual todos poseen acceso y mejorará la persistencia de datos y no ocupará tanta memoria en el dispositivo.

En caso de no existir ningún vídeo en la llamada, se mostraría un texto descriptivo donde quedase constancia de que no hay ningún vídeo para mostrar.

Para poder llamar al servicio, en este caso al sí poder programarlo, se ha vuelto a emplear el recurso de los *providers*. Se ha creado un provider para gestionar los dos tipos de llamadas que se van a utilizar en nuestro código:

- **getPlaylistsForChannel(channel):** llamada al servicio que nos devolverá los playlist del canal que deseemos ver. En este caso al tener el canal y la playlist concreta, la primera, podemos fijar el identificador del canal.
- **getListVideos(listId):** llamada al servicio que nos devolverá todos los vídeos del id de la playlist que se le pase como parámetro, en este caso será el primer argumento de la lista de vídeos que haya creado la llamada anterior.

Observar un tutorial

No se quiere obligar al usuario a tener que abandonar nuestra aplicación para tener que visualizar el tutorial. Para evitarlo, cuando el usuario pulsa en alguno de los vídeos para poder verlo, se redirige a una nueva ventana que se pasa como parámetro el vídeo seleccionado.

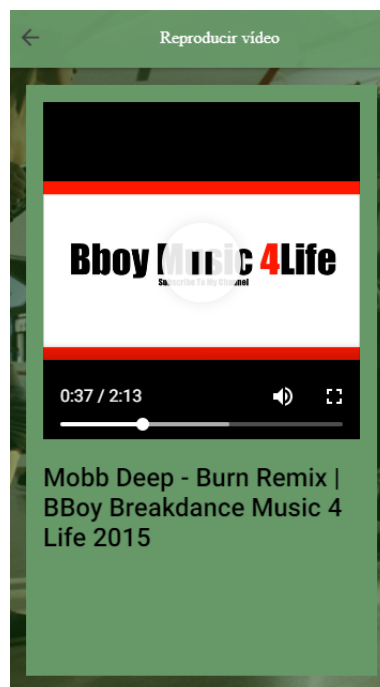
En esta nueva ventana se muestra un *iframe*, el cual es *responsive*, donde muestra el vídeo a reproducir desde YouTube y de esta forma podrá visualizarlo en nuestra aplicación, pero con las mismas funcionalidades: pantalla completa, ajustar volumen, velocidad, etc.

Para poder volver, posee un botón al margen superior derecho (siguiendo la estructura de la aplicación) para poder volver a la pantalla de tutoriales.

Como ya se propuso en el prototipo, el fondo es diferente y la barra de navegación desaparece para dar constancia de que está a un nivel diferente de navegación y tenga un mayor control del espacio.

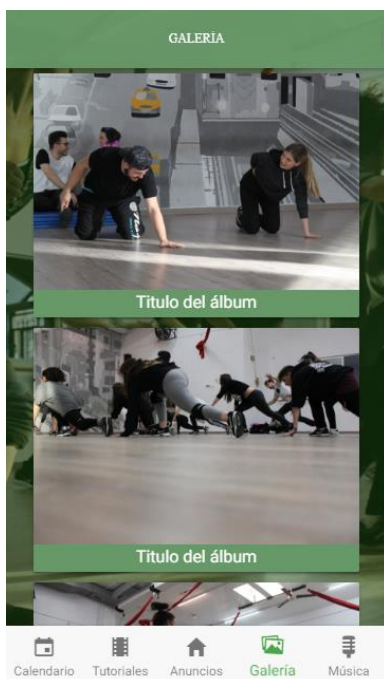
El contenido abarca casi toda la vista para dar mayor énfasis y que no distraiga la atención del usuario a otros elementos.

En este caso, como se accede a una api externa de la cual no se posee la capacidad de modificar, y al tratarse de un vídeo puede darse el caso de que tarde unos segundos, se ha implementado, gracias a la librería *Loading* que ofrece angular, una vista que muestra una animación mientras se carga el contenido de la lista, para dar información del proceso y que no de margen a pensar un error, si no de constancia de que el contenido está siendo cargado.



5.7 Galería

Para esta funcionalidad se ha decidido prescindir del método de cuadrícula ya que podía llevar a fallos de diseño por el redimensionamiento de las imágenes, por ello se ha establecido una estructura de una tabla donde se muestra primero una imagen resumen y seguidamente el título del álbum a mostrar.



Estos datos se obtendrán realizando una petición GET a la API correspondiente. La respuesta será enviada en .json con los diferentes álbumes que hayan subidos a la nube. Dentro de un álbum, a parte del identificador, el nombre y la imagen resumen, tendrá un array de imágenes.

En la primera vista tan solo mostraremos los álbumes. Al no tener el servicio creado, se han moqueado doce fotos para poder hacer un testeo del funcionamiento de la aplicación, estas fotos están contenidas en la carpeta “assets/img” del proyecto.

Al pinchar sobre cualquiera de las imágenes, se redireccionará a una nueva ventana. Esta nueva ventana inicializará un objeto con los datos obtenidos pasado por los parámetros del controlador que nos ofrece Ionic.

En esta nueva ventana llamada Álbum se muestran ahora si todas las imágenes contenidas en el álbum seleccionado, pero sin ningún tipo de título descriptivo. El usuario puede navegar por el contenido mediante un simple gesto con el dedo para deslizar la pantalla hacia arriba o hacia abajo. Está preparado para poder ser redimensionado por si decide apaisar la pantalla del dispositivo.

Lo único complicado de esta nueva ventana ha sido implementar la funcionalidad que permite al usuario poder subir nuevas imágenes. Mediante un botón, se realiza una función donde se abre la galería del terminal, selecciona una imagen, y al terminar esa acción se realiza un método POST a la API la cual gestionará esta información para poder subir la imagen.

Al recibir la respuesta, en caso de ser aceptada, se actualizaría el álbum mostrando esta nueva imagen, en su defecto, saltaría una alerta aclarando al usuario que no ha sido posible satisfacer su acción.

Una vez abierta la imagen, se han implementado un total de cuatro funciones para gestionar la imagen abierta:

- `shareInstagram()`: Gracias a un plugin ofrecido por ionic native (instagram) en este método se crea una instancia a un objeto que nos ofrece el plugin donde enviamos la imagen para poder compartirla con esta plataforma.

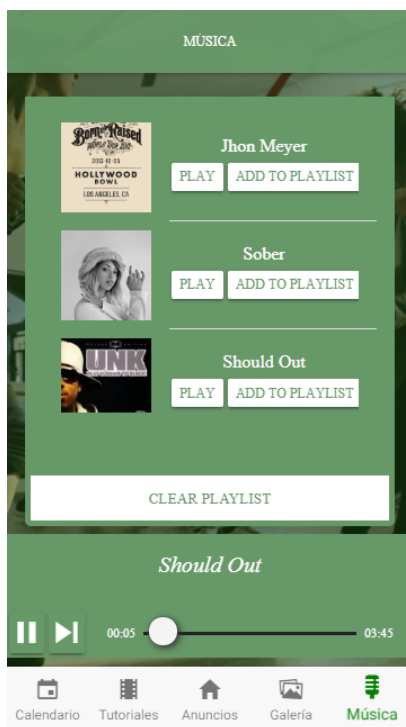




- `download()`: Haciendo uso de otro plugin nativo (Base64ToGallery) se transforma la imagen que se está mostrando a un formato compatible y se guarda en la gallería del teléfono, descargando de esta manera la imagen del servidor.
- `next()` y `back()`: estos métodos serán para navegar desde esta ventana por todas las imágenes del álbum seleccionado por si se desea este modo de navegación. Actualmente al tratarse de imágenes guardadas y accedidas de un directorio no posee la funcionalidad que se espera.

5.8 Música

En esta funcionalidad se han realizado ligeros cambios del diseño. Se evaluó si existía realmente la necesidad de implementar la funcionalidad que si pulsaba una canción se abriera en forma de reproductor ya que toda la información demás que pudiera añadir esa pantalla eran tan solo mostrar más grande el texto que ya se mostraba en la ventana anterior, en cambio sumaba las posibilidades de un posible error, por lo que se decidió prescindir de ella, y suplir sus objetivos en una sola ventana.



Para conseguirlo se ha dividido la vista en dos, una lista fija donde se muestran las canciones, donde ya se muestra un pequeño icono de la canción, el título de la canción (no se ha plasmado más información del contenido multimedia puesto que se pudo ver que no era necesario para los clientes) y dos botones: reproducir y lista de reproducción. Y una segunda vista fija en la parte inferior donde se muestra un reproductor vacío que se actualizará cuando una de las canciones esté sonando o, en su defecto, cuando haya sido pausada.

Empezando por la clase, se supondrá que llamaremos a un servicio donde nos devolverá un listado en un archivo .json con los datos en la estructura deseada, para mapearlos y posteriormente mostrarlos. Al no poseer actualmente esta llamada, se han moqueado los datos para poder probar nuestro dispositivo con música real.

Disponemos de dos listas, una donde posee todas las canciones y otra donde se irá actualizando según el usuario vaya añadiendo canciones a la lista de reproducción. A su vez disponemos de dos variables, una donde tendremos guardada la canción actual y su índice.

Con estas variables, podremos gestionar mediante condicionales las posiciones de la próxima canción que debe sonar, primero comprobaremos si hay *playlist*, en caso de no haberla simplemente reproducirá la siguiente hasta llegar al final, que, en ese caso, volverá al principio. En caso de que exista una *playlist*, seguirá la misma lógica, pero con el contenido de la lista.

Puede darse el caso de que el usuario se equivoque y quiera borrar el contenido de la lista de reproducción, por ello se ha implementado un botón para poder resetear por completo dicha lista y poder volver a empezar de nuevo. Esta lista no es persistente, cada vez que se inicializa la vista, se limpiará.

En cuanto al reproductor, se ha establecido dos botones (pausa y reproducir) los cuales solo se mostrará uno de ellos según el estado de nuestra canción actual (en reproducción o en pausa). Para hacer posible esto, se ha utilizado de otra librería nativa de Ionic; ionic-audio. Esta ofrece un objeto que permite ser tratado como una canción.

En este caso, lo complejo del código no está en el controlador, si no en el archivo .html del directorio. Para poder reproducir la canción se utiliza del recurso que ofrece la plataforma: ng y audio-track.

En cuanto a la usabilidad, se ha decidido mostrar el reproductor de primeras con el fin de que el usuario sepa que si pulsa alguna de las canciones se reproducirá en esa pantalla y no abrirá una nueva. A su vez, otro detalle a tener en cuenta, es que si el usuario reproduce una canción puede navegar por el resto de la aplicación sin que esta se detenga.

5.9 Clean Code²³

Para un buen programador debe ser fundamental poder crear un código que pueda ser transferible a otros programadores, e incluso poder facilitarle a sí mismo la accesibilidad de sus propios métodos. Por ello, toda la implementación se ha intentado crear siempre con los mismos patrones, intentando saturar de código las clases y subdividiendo las tareas lo máximo posible.

A su vez, las propias clases siempre se estructuran de la siguiente forma:

- Imports necesarios
- Clases auxiliares si fueran necesarias
- Información de la página
- Variables de la clase principal
- Constructor
- Métodos por orden de llamada

A su vez todo el código está implementado en inglés. Esto también ayuda a entender mucho más fácil los errores. La mayoría de documentación oficial sobre los lenguajes usados está en inglés, y no solo eso, también los hilos creados en diferentes páginas (*git*, *stackoverflow*, etc.) están redactados en inglés, por lo que tener un código implementado en este idioma, facilita mucho a la hora de encontrar las similitudes con esta información y ahorrar tiempo de programación en estos posibles fallos.

²³ “[Robert C. Martin](#) – (2008) Clean Code: A Handbook of Agile Software Craftsmanship”

Otro punto a tener en cuenta, meramente estético, es que los métodos que ofrecen los plugin de *ionic native* están todos en inglés, por lo que tener en una misma clase métodos en inglés y castellano muestra falta de experiencia en el sector.

6. Pruebas

Con el objetivo de poder testear nuestra aplicación y comprobar su correcto funcionamiento, se ha creado un diccionario de pruebas que cumplen todas las posibles funcionalidades de nuestra aplicación para poder comprobar su correcto funcionamiento.

Al no poseer la totalidad de la API a la que se hace referencia, varias funcionalidades no pueden ser testeadas ya que los datos han sido maquetados para comprobar el funcionamiento y respuesta de la aplicación.

6.1 Diseño de Pruebas

6.1.1 Vista Login

Nombre	Validación de los campos
Propósito	Verificar que los campos que se introducen son correctos
Prerrequisitos	Que los campos hayan sido rellenados con la información que se espera: email y contraseña de algún usuario creado.
Entrada	<ul style="list-style-type: none">● Rellenar los campos email y contraseña y pulsar el botón acceder.
Salida esperada	Push a la ventana de anuncios

Nombre	Registro de Usuario
Propósito	Verificar que los campos que se introducen son correctos
Prerrequisitos	Que los campos hayan sido rellenados con la información que se espera: email válido y contraseña válida.
Entrada	<ul style="list-style-type: none">● Rellenar los campos email y contraseña y pulsar el botón registrar.
Salida esperada	Alerta “Usuario creado con éxito” y push a la ventana inicial.

6.1.2 Vista Anuncios

Nombre	Crear una nueva lista de asistencia
Propósito	Push a listas de asistencia y llamada al servicio para que aparezcan los nombres de los miembros de la compañía.
Prerrequisitos	Que el dispositivo posea internet.
Entrada	<ul style="list-style-type: none">● Pulsar el botón de la cabecera.
Salida esperada	Push a nueva ventana, mostrar listado con los miembros de la compañía.

Nombre	Mostrar últimos eventos
Propósito	Get al servicio para que devuelva los eventos cercanos a la fecha actual.
Prerrequisitos	Que el dispositivo posea internet. Fase de autenticación con éxito.
Entrada	-
Salida esperada	Actualización de la tabla mostrando los últimos eventos creados.

6.1.3 Calendario

Nombre	Mostrar calendario actualizado
Propósito	Mostrar y actualizar el calendario al día de hoy mostrando, si lo hay, eventos creados
Prerrequisitos	Estar en la pantalla de calendario.
Entrada	-
Salida esperada	Actualizar los eventos y mostrarlos en el calendario diseñado.

Nombre	Añadir Evento
Propósito	Crear un nuevo evento y mostrarlo en el calendario
Prerrequisitos	Estar en la pantalla de calendario. Pulsar el botón de la cabecera Tener internet
Entrada	Rellenar el formulario: -titulo

	<ul style="list-style-type: none"> - lugar - hora - fecha - pulsar Crear
Salida esperada	PopUp mostrando información de la acción, push a la ventana de calendario, actualización del calendario.

Nombre	Eliminar Evento
Propósito	Eliminar un evento del calendario
Prerrequisitos	Estar en la pantalla de calendario. Haber seleccionado un día que posea evento.
Entrada	<ul style="list-style-type: none"> - Un evento - Pulsar el botón eliminar del evento
Salida esperada	Borrar elemento de la API, del calendario y un popUp mostrando el resultado de la acción.

Nombre	Editar Evento
Propósito	Editar y actualizar un evento ya creado.
Prerrequisitos	Estar en la pantalla de calendario. Haber seleccionado un día que posea evento y pulsar el botón editar.
Entrada	<ul style="list-style-type: none"> - Un evento - Rellenar el formulario con lo que se desea cambiar. - Pulsar el botón actualizar.
Salida esperada	Actualizar elemento de la API, del calendario, un popUp mostrando el resultado de la acción, volver a la pantalla de calendario.

Nombre	Mostrar Evento
Propósito	Mostrar contenido de un día que posea eventos
Prerrequisitos	Estar en la pantalla de calendario. Haber seleccionado un día que posea evento.
Entrada	-
Salida esperada	Nuevo contenedor con la información del/los evento/s que existen en el día seleccionado.

6.1.4 Galería

Nombre	Mostrar álbumes
Propósito	Mostrar todos los álbumes que hay en el servicio
Prerrequisitos	Estar en la pantalla de galería.
Entrada	- Id del usuario
Salida esperada	Mostrar por pantalla todos los álbumes y sus títulos correspondientes.

Nombre	Eliminar álbum
Propósito	Eliminar álbum ya creado.
Prerrequisitos	Estar en la pantalla de galería, abrir un álbum y presionar el botón eliminar.
Entrada	- Un álbum
Salida esperada	Eliminar de la API el álbum, mostrar popUp, volver a la pantalla de galería, actualizar eliminando el álbum borrado.

Nombre	Editar álbum
Propósito	Editar álbum ya creado.
Prerrequisitos	Estar en la pantalla de galería, abrir un álbum y presionar el botón editar.
Entrada	- Cambiar el título y pulsar botón editar.
Salida esperada	Editar en la api el álbum, mostrar popUp, volver a la pantalla del álbum, actualizar el título.

Nombre	Añadir imagen
Propósito	Añadir una nueva imagen a un álbum
Prerrequisitos	Estar en la pantalla de un álbum y presionar añadir imagen.
Entrada	- Una imagen
Salida esperada	Redirigir al usuario a su galería, subir la imagen a la API y actualizar el contenido del álbum.

Nombre	Mostrar Imágenes
Propósito	Mostrar todas las imágenes de un álbum
Prerrequisitos	Estar en la pantalla de galería y pulsar sobre un álbum.
Entrada	-
Salida esperada	Llamada a la API y mostrar todas las imágenes del álbum seleccionado.

Nombre	Mostrar Imagen
Propósito	Mostrar en grande una sola imagen
Prerrequisitos	Estar sobre un álbum y presionar una sola imagen.
Entrada	- Touch sobre una imagen
Salida esperada	Redirigir a una nueva ventana con diversos botones y la imagen en grande.

Desarrollo Centrado en el Usuario para la construcción de una aplicación móvil de gestión de una compañía de danza (Quakes)

Nombre	Compartir Imagen en Instagram
Propósito	Compartir imagen seleccionada en Instagram.
Prerrequisitos	Haber mostrado una imagen, seleccionar el botón compartir en Instagram, tener internet y cuenta de Instagram.
Entrada	- Una imagen
Salida esperada	Redirigir al usuario a su cuenta de Instagram con la imagen cargada y lista para publicar.

Nombre	Eliminar Imagen
Propósito	Eliminar una imagen de un álbum
Prerrequisitos	Haber mostrado una imagen, seleccionar el botón eliminar.
Entrada	- Una imagen. -
Salida esperada	Eliminar de la API la imagen, mostrar popUp, volver a la pantalla del álbum, actualizar eliminando la imagen borrada.

Nombre	Descargar Imagen
Propósito	Guarda una imagen de un álbum en el terminal.
Prerrequisitos	Haber mostrado una imagen, seleccionar el botón descargar.
Entrada	- Una imagen
Salida esperada	Descargar la imagen en el terminal.

6.1.5 Tutoriales

Nombre	Mostrar Tutoriales
Propósito	Mostrar los tutoriales disponibles en la playlist de YouTube
Prerrequisitos	Tener Internet, estar en la pantalla de tutoriales.
Entrada	-
Salida esperada	Un listado con todos los vídeos disponibles para reproducir.

Nombre	Reproducir un Tutorial
Propósito	Reproducir un tutorial
Prerrequisitos	Seleccionar uno de los vídeos mostrados en la pantalla de tutoriales.
Entrada	-
Salida esperada	Redireccionamiento a una nueva venta. Mostrar el vídeo en grande con la posibilidad de poder reproducirlo.

6.1.6 Reproductor de Música

Nombre	Mostrar Música
Propósito	Mostar un listado de las canciones disponibles en el servidor
Prerrequisitos	Estar en la pantalla de música
Entrada	-
Salida esperada	Un listado de las canciones que se pueden reproducir.

Nombre	Reproducir una canción
Propósito	Reproducir una canción en nuestro reproductor.
Prerrequisitos	Pulsar play sobre una de las canciones
Entrada	- Una canción
Salida esperada	Actualizar el reproductor de la aplicación y comenzar a reproducir la canción seleccionada.

Desarrollo Centrado en el Usuario para la construcción de una aplicación móvil de gestión de una compañía de danza (Quakes)

Nombre	Pausar una canción
Propósito	Pausar una canción que esté reproduciéndose.
Prerrequisitos	Que haya una canción en curso. Pulsar el botón pause.
Entrada	- Acción sobre el botón “pause”
Salida esperada	Pausar la canción en curso.

Nombre	Añadir a una playlist
Propósito	Añadir una o varias canciones a una playlist temporal.
Prerrequisitos	Pulsar sobre el botón add to playlist
Entrada	- Una canción
Salida esperada	Una playlist temporal para reproducir.

Nombre	Limpiar Playlist
Propósito	Eliminar las canciones añadidas a la playlist temporal
Prerrequisitos	Pulsar sobre el botón Clear Playlist y que exista una playlist
Entrada	- Una playlist
Salida esperada	PopUp informando que la playlist ha sido eliminada.

Nombre	Reproducir siguiente canción
Propósito	Reproducir la siguiente canción de la playlist y en caso de que no haya reproducir en orden de listado.
Prerrequisitos	Haber seleccionad una canción antes, que exista una playlist o que haya una canción tras la seleccionada en la lista.
Entrada	-
Salida esperada	Reproducir la siguiente canción. En caso de existir playlist tendrá prioridad.

7. Conclusión

Tras un proceso consistente, se ha logrado crear una aplicación para solventar un problema real, donde el alumno ha tenido que superar nuevos retos: tratar con usuarios reales, utilizar plataformas y lenguajes nuevos no enseñados en el grado, crear diversos prototipos y solventar contratiempos que proporcionan las diferentes plataformas usadas en términos de incompatibilidades, problemas de versiones, etc.

Se ha logrado poder testear la aplicación con los clientes reales teniendo un *feedback* bastante positivo, donde se puede mejorar, implementar más funcionalidades, limar el código, pero con un resultado ya positivo para el caso a solventar.

La metodología empleada ha ayudado a cubrir un problema creando lo justo y necesario, sin tener que crear funcionalidades que hubieran acabado siendo inútiles, malgastando tiempo, que, en caso de tratarse de un contenido manufacturado, hubiera supuesto un uso de horas y dinero malgastados.

En resumen, el trabajo de fin de grado ha supuesto una evolución al alumno donde ha aprendido de manera autodidacta a solventar problemas que en un futuro trabajo le resultarán similares. Ha logrado aprender un nuevo lenguaje y los problemas que conlleva. Ha creado una plataforma que futuramente podría llegar a ser inclusive comercializada y ha supuesto una consolidación de muchos conocimientos adquiridos en el grado.

7.1 Relación del trabajo desarrollado con los estudios cursados

En cuanto a la relación con el grado, gracias a los conocimientos adquiridos ha resultado más sencillo la comprensión de un nuevo lenguaje orientado a objetos. Se ha aplicado el conocimiento web adquirido en cuanto al etiquetado html.

Todo el trabajo ha sido orientado en la metodología propuesta en una asignatura concreta de la rama de tecnologías de la información, donde abarca varios puntos interesantes, y obliga al alumno a entender que programar no solo consiste en crear código, si no que hay que orientarlo a un objetivo final, que en este caso es el usuario.

A su vez se ha familiarizado con las llamadas (get, push, put, delete.) a un servicio externo e incluso se ha gestionado la compatibilidad con una API ya creada de una plataforma de grandes dimensiones (YouTube).

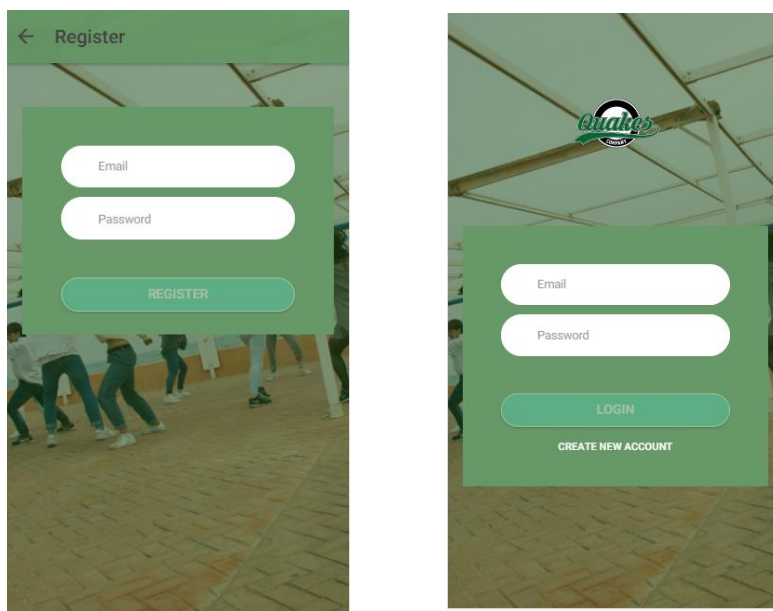
Por último, aunque no mostrado en la memoria, se ha llevado a cabo el proyecto con una gestión propia en trello con un sistema de tableros para llevar una consistencia de lo que esta pendiente de realizar, las tareas en progreso y las tareas ya terminadas.

Referencias

1. <https://ionicframework.com/>
2. <https://angular.io/>
3. <http://xtremefunctional.com/>
4. <https://www.freeletics.com/es>
5. <https://play.google.com/store/apps/details?id=com.proyecto.atalantasportclub.tgcustom&hl=es>
6. <https://play.google.com/store/apps/details?id=com.proyecto.activaclub.tgcustom&hl=es>
7. <https://play.google.com/store/apps/details?id=com.semicinternet.danza>
8. <https://play.google.com/store/apps/details?id=com.fitnessmobileapps.millenniumdancecomplex>
9. “Gracia Bandrés, M.A., Gracia Murugarren, J., Romero San Martín, D. – (2015) TecsMedia: Metodologías de diseño centradas en usuarios”
10. <https://www.google.es/intl/es/forms/about/>
11. Y. Hassan, F.J. Martín, G. Iazza (2004): Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información.
https://www.upf.edu/hipertextnet/numero-2/disenio_web.html
12. <http://mpiua.invid.udl.cat/bocetos-sketching/>
13. <https://balsamiq.cloud/>
14. A. Floriá Cortes (2000): Recopilación de métodos de usabilidad SIDAR Di ibl bilidad. SIDAR.
<http://www.sidar.org/recur/desdi/traduc/es/visitable/inspection.htm#Cogn>
15. <http://www.sidar.org/recur/desdi/traduc/es/visitable/testing.htm>
16. Hassan Montero, Y.; Martín Fernández, F.J.; Hassan Montero, D.; Martín Rodríguez, O. Arquitectura de la Información en los entornos virtuales de aprendizaje: Aplicación de la técnica de Card Sorting y análisis cuantitativo de los resultados.
17. <https://www.justinmind.com/>
18. <https://ionicframework.com/docs/components/#cards>
19. “Jakob Nielsen – (2000) USABILIDAD, DISEÑO DE SITIOS WEB”
20. <https://angular.io/guide/typescript-configuration>
21. <https://ionicframework.com/docs/native/>
22. <https://ionicframework.com/docs/components/>
23. “Robert C. Martin – (2008) Clean Code: A Handbook of Agile Software Craftsmanship”

Apéndice

Apéndice I – Manual de Usuario



Pantalla de “**Login**” donde se puede realizar las siguientes acciones:

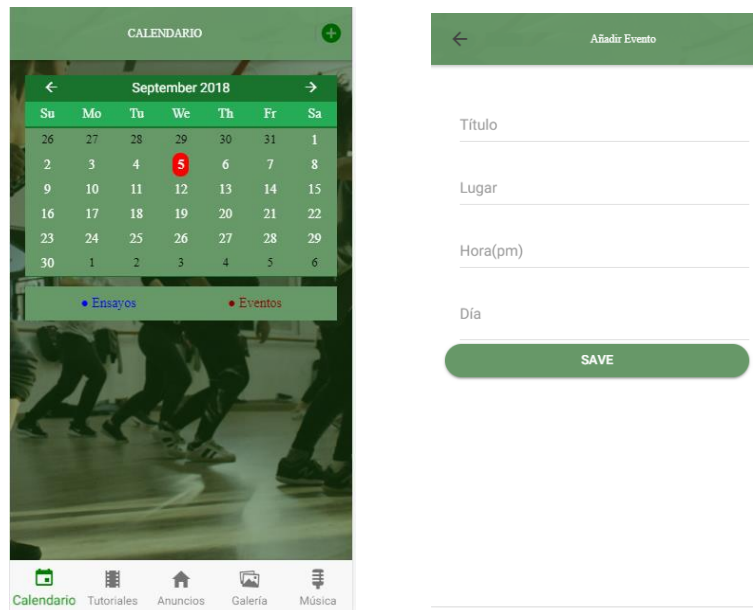
- Acceder con un email y un password
- Crear una nueva cuenta



Pantalla de “**Home**” donde se puede realizar las siguientes acciones:

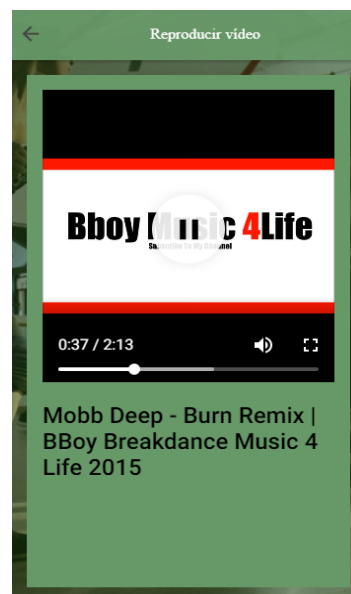
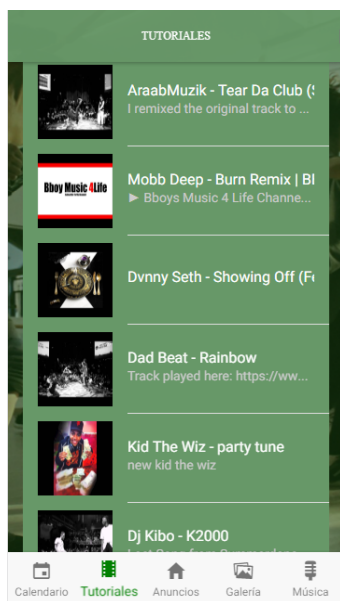
- Mostrar Listas de asistencia
- Observar mediante scroll los eventos de hoy y los próximos eventos

Desarrollo Centrado en el Usuario para la construcción de una aplicación móvil de gestión de una compañía de danza (Quakes)



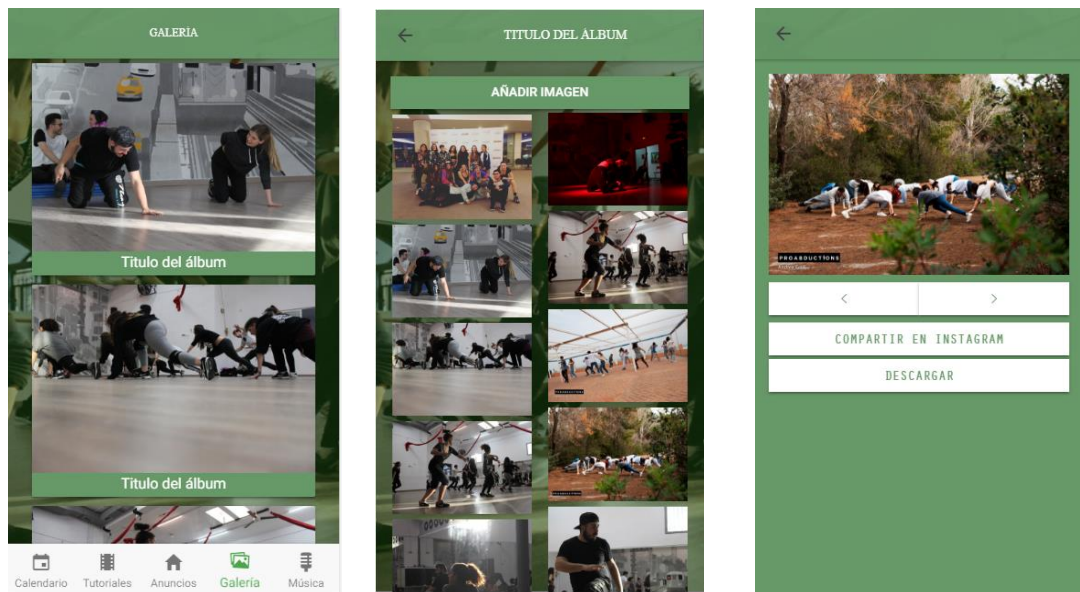
Pantalla de “**Calendario**” donde se puede realizar las siguientes acciones:

- Mostrar un evento de un día concreto
- Interactuar con el calendario para buscar una fecha concreta
- Eliminar un evento
- Editar un evento
- Crear un nuevo evento con el botón de la cabecera.



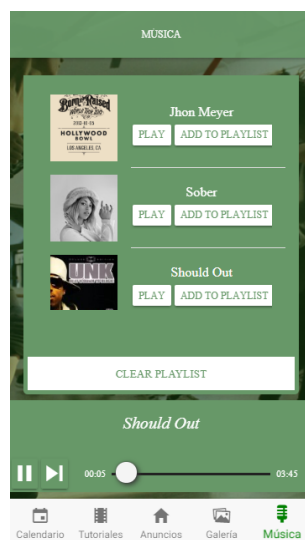
Pantalla de “**Tutoriales**” donde se puede realizar las siguientes acciones:

- Mostrar listado de tutoriales disponibles
- Poder ver un tutorial
- Interactuar con el vídeo



Pantalla de “**Galería**” donde se puede realizar las siguientes acciones:

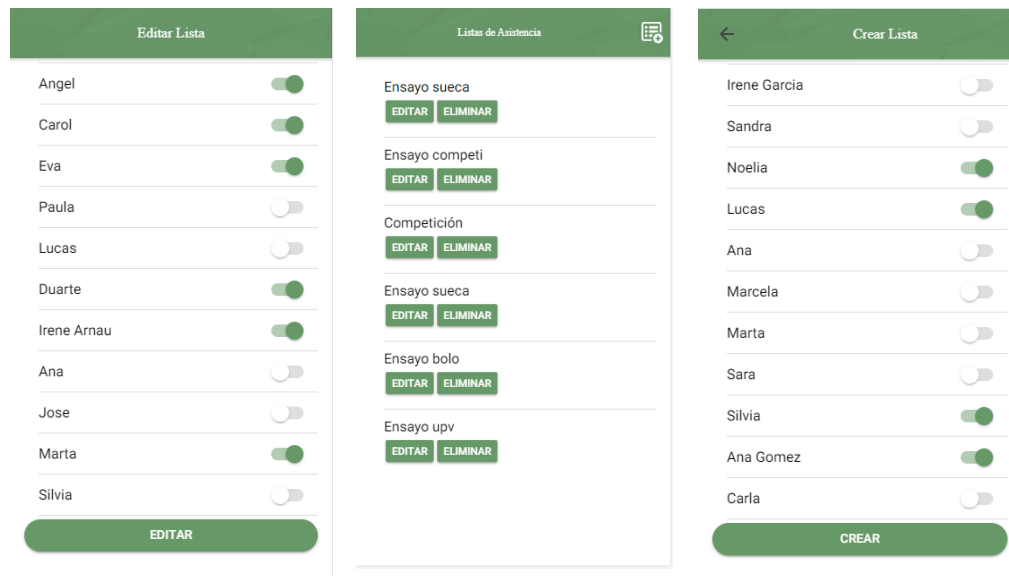
- Mostrar listado de álbumes disponibles
- Seleccionar un álbum
- Editar o Eliminar un álbum
- Añadir imágenes a un álbum
- Observar una imagen en concreto
- Compartir esa imagen en Instagram
- Guardar la imagen seleccionada en el terminal
- Navegar entre las imágenes del álbum seleccionado.



Pantalla de “**Música**” donde se puede realizar las siguientes acciones:

- Mostrar listado de músicas disponibles para reproducir
- Reproducir una música
- Navegar en la canción actual
- Limpiar o Añadir a una playlist temporal.
- Navegar, pausar y reanudar las canciones mostradas en la lista.

Desarrollo Centrado en el Usuario para la construcción de una aplicación móvil de gestión de una compañía de danza (Quakes)



Pantalla de “**Listas de Asistencia**” donde se puede realizar las siguientes acciones:

- Mostrar listas de asistencia disponibles
- Eliminar una lista ya creada
- Editar una lista ya creada donde se muestra la información de la lista
- Crear una nueva lista
- Eliminar algún miembro a la hora de crear una nueva lista